SOFTIMAGE®|XSI®
Version 5.0
Animation
SOFTIMAGE|XSI includes the Python Release 2.3.2 software. The Python software is:
Copyright © 2001, 2002, 2003 Python Software Foundation. All rights reserved.
Copyright © 2000 BeOpen.com. All rights reserved.
Copyright © 1995-2000 Corporation for National Research Initiatives. All rights reserved.
Copyright © 1991-1995 Stichting Mathematisch Centrum. All rights reserved.

SOFTIMAGE|XSI uses the OpenEXR software © 2002, Industrial Light & Magic, a division of Lucas Digital Ltd. LLC All rights reserved. Redistribution and use in source and binary forms, with
without modification, are permitted provided that the following conditions are met: [1]
Redistributions of source code must retain the above copyright notice, this list of conditions and
the following disclaimer. [2] Redistributions in binary form must reproduce the above copyright
notice, this list of conditions and the following disclaimer in the documentation and/or other
materials provided with the distribution. [3] Neither the name of Industrial Light & Magic nor the
names of its contributors may be used to endorse or promote products derived from this software
without specific prior written permission.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO,
THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR
PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR
CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO,
PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR
PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF
LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING
NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS
SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

SOFTIMAGE|XSI includes the libtiff library. The libtiff software is:
Copyright (c) 1988-1997 Sam Leffler
Copyright (c) 1991-1997 Silicon Graphics, Inc.
Permission to use, copy, modify, distribute, and sell this software and its documentation for any
purpose is hereby granted without fee, provided that (i) the above copyright notices and this
permission notice appear in all copies of the software and related documentation, and (ii) the
names of Sam Leffler and Silicon Graphics may not be used in any advertising or publicity relating
to the software without the specific, prior written permission of Sam Leffler and Silicon Graphics.

THE SOFTWARE IS PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND,
EXPRESS, IMPLIED OR OTHERWISE, INCLUDING WITHOUT LIMITATION, ANY
WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.
IN NO EVENT SHALL SAM LEFFLER OR SILICON GRAPHICS BE LIABLE FOR ANY
SPECIAL, INCIDENTAL, INDIRECT OR CONSEQUENTIAL DAMAGES OF ANY KIND, OR
ANY DAMAGES WHATSOEVER RESULTING FROM LOSS OF USE, DATA OR PROFITS,
WHETHER OR NOT ADVISED OF THE POSSIBILITY OF DAMAGE, AND ON ANY THEORY
OF LIABILITY, ARISING OUT OF OR IN CONNECTION WITH THE USE OR
PERFORMANCE OF THIS SOFTWARE.

SOFTIMAGE|XSI includes the libpng-1.2.5 library. The libpng software is:
Copyright (c) 2000-2002 Glenn Randers-Pehrson, and is distributed according to the same
disclaimer and license as libpng-1.0.6 with the following individuals added to the list of
Contributing Authors: Simon-Pierre Cadieux, Eric S. Raymond, and Gilles Vollant.
There is no warranty against interference with your enjoyment of the library or against
infringement. There is no warranty that our efforts or the library will fulfill any of your particular
purposes or needs. This library is provided with all faults, and the entire risk of satisfactory quality,
performance, accuracy, and effort is with the user.

THE PNG REFERENCE LIBRARY IS SUPPLIED "AS IS". THE CONTRIBUTING AUTHORS AND GROUP 42,
INC. DISCLAIM ALL WARRANTIES, EXPRESSED OR IMPLIED, INCLUDING, WITHOUT LIMITATION,
THE WARRANTIES OF MERCHANTABILITY AND OF FITNESS FOR ANY PURPOSE. THE CONTRIBUTING
AUTHORS AND GROUP 42, INC. ASSUME NO LIABILITY FOR DIRECT, INDIRECT, INCIDENTAL, SPECIAL,
EXEMPLARY, OR CONSEQUENTIAL DAMAGES, WHICH MAY RESULT FROM THE USE OF THE PNG
REFERENCE LIBRARY, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

SOFTIMAGE|XSI includes the libjpeg library. This software is based in part on the work of the
Independent JPEG Group. The libjpeg library is:
Copyright 1991-1998, Thomas G. Lane. All Rights reserved.
Permission is hereby granted to use, copy, modify, and distribute this software (or portions
thereof) for any purpose, without fee, subject to these conditions: [1] If any part of the source code
for this software is distributed, then this README file must be included, with this copyright and
no-warranty notice unaltered; and any additions, deletions, or changes to the original files must be
clearly indicated in accompanying documentation. [2] If only executable code is distributed, then
the accompanying documentation must state that "this software is based in part on the work of the
Independent JPEG Group". [3] Permission for use of this software is granted only if the user
accepts full responsibility for any undesirable consequences; the authors accept NO LIABILITY for damages of any kind.

SOFTIMAGE|XSI Mod Tool includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/). The OpenSSL toolkit is:

Copyright (c) 1998-2003 The OpenSSL Project. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met: [1.] Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer. [2.] Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. [3.] All advertising materials mentioning features or use of this software must display the following acknowledgment: "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (http://www.openssl.org/)" [4.] The names "OpenSSL Toolkit" and "OpenSSL Project" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact openssl-core@openssl.org. [5.] Products derived from this software may not be called 'OpenSSL' nor may 'OpenSSL' appear in their names without prior written permission of the OpenSSL Project. [6.] Redistributions of any form whatsoever must retain the following acknowledgment: "This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit. (http://www.openssl.org/)". THIS SOFTWARE IS PROVIDED BY THE OPENSSL PROJECT "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE OpenSSL PROJECT OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).

The SSLeay library is:

Copyright (C) 1995-1998 Eric Young (eay@cryptsoft.com) All rights reserved.

This package is an SSL implementation written by Eric Young (eay@cryptsoft.com). The implementation was written so as to conform with Netscape SSL.

This library is free for commercial and non-commercial use as long as the following conditions are adhered to. The following conditions apply to all code found in this distribution, be it the RC4, RSA, lhash, DES, etc., code; not just the SSL code. The SSL documentation included with this distribution is covered by the same copyright terms except that the holder is Tim Hudson (tjh@cryptsoft.com).

Copyright remains Eric Young's, and as such any Copyright notices in the code are not to be removed. If this package is used in a product, Eric Young should be given attribution as the author of the parts of the library used. This can be in the form of a textual message at program startup or in documentation (online or textual) provided with the package.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

[1.] Redistributions of source code must retain the copyright notice, this list of conditions and the following disclaimer. [2.] Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution. [3.] All advertising materials mentioning features or use of this software must display the following acknowledgment: "This product includes cryptographic software written by Eric Young (eay@cryptsoft.com)". The word 'cryptographic' can be left out if the routines from the library being used are not cryptographic related. [4.] If you include any Windows specific code (or a derivative thereof) from the apps directory (application code) you must include an acknowledgement: "This product includes software written by Tim Hudson (tjh@cryptsoft.com)"

THIS SOFTWARE IS PROVIDED BY ERIC YOUNG "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE AUTHOR OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY
OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The licence and distribution terms for any publically available version or derivative of this code cannot be changed. i.e. this code cannot simply be copied and put under another distribution licence (including the GNU Public Licence.)

SOFTIMAGE|XSI includes software developed by the Apache Software Foundation (http://www.apache.org/). The Xerces software is:

Copyright (c) 1999-2003 The Apache Software Foundation. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The end-user documentation included with the redistribution, if any, must include the following acknowledgment: "This product includes software developed by the Apache Software Foundation (http://www.apache.org/)." Alternately, this acknowledgment may appear in the software itself, if and wherever such third-party acknowledgments normally appear.
4. The names "Xerces" and "Apache Software Foundation" must not be used to endorse or promote products derived from this software without prior written permission. For written permission, please contact apache@apache.org.
5. Products derived from this software may not be called "Apache", nor may "Apache" appear in their name, without prior written permission of the Apache Software Foundation.

THIS SOFTWARE IS PROVIDED "AS IS" AND ANY EXPRESSED OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE APACHE SOFTWARE FOUNDATION OR ITS CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

This software consists of voluntary contributions made by many individuals on behalf of the Apache Software Foundation and was originally based on software copyright (c) 1999, International Business Machines, Inc., http://www.ibm.com. For more information on the Apache Software Foundation, please see http://www.apache.org/.

This document is protected under copyright law. The contents of this document may not be copied or duplicated in any form, in whole or in part, without the express written permission of Avid Technology, Inc. This document is supplied as a guide for the Softimage product. Reasonable care has been taken in preparing the information it contains. However, this document may contain omissions, technical inaccuracies, or typographical errors. Avid Technology, Inc. does not accept responsibility of any kind for customers' losses due to the use of this document. Product specifications are subject to change without notice.

Documentation Team
Judy Bayne, Stephen Blair, Grahame Fuller, Amy Green, Edna Kruger, Luc Langevin, and Jamal Rahal.

09 2005
Contents

Animation

Copyright and Disclaimer ........................................ 2

Animation

Chapter 1

What Can You Animate in XSI? .................................. 19
How Do You Animate in XSI? .................................. 19
Levels of Animation ........................................ 21
Order of Animation ........................................ 22
Tools for Animation ........................................ 24
Getting Animation Commands ................................. 24
General Tools for Animation .................................. 25
Playing the Animation ......................................... 27
Getting Animation Information ............................... 28
Setting Animation Preferences ............................... 28
Selecting Types of Animated Objects ....................... 29
Finding Animation Elements .................................. 30
In the Explorer ........................................ 30
In the Schematic View ......................................... 31
In the Spreadsheet ........................................ 32

Chapter 2

Tasks for Any Type of Animation .............................. 33

Marking Parameters for Animation .......................... 35
Setting a Preference for Marked Parameters ............... 35
Marking Parameters ........................................ 36
Marking Custom Parameters and Sets ..................... 38
Locking Marked Parameters .................................. 38
Clearing (Unmarking) Marked Parameters ................. 39
Remembering Marked Parameters with Marking Sets ...... 39
Locking Animation for Objects and Parameters .......... 41
Ghosting Animated Objects .................................. 42
Activating Ghosting for Objects ............................. 43
Activating Ghosting for Layers and Groups ............... 43
Activating Ghosting in 3D Views ............................ 44
Controlling the Ghost Display .............................. 45
Selecting the Ghosting Display Type ....................... 49
Copying Animation .......................................... 52
Using the Animation Menu .................................. 52
Using the Explorer or a Property Editor .................. 53
Copying Animation between Models and Hierarchies ...... 54
Scaling and Offsetting (Retiming) Animation ............... 55
Offsetting or Scaling Animation ............................ 56
## Chapter 4: Animating with Keys

- **Viewing Keys** ........................................ 102
- **Setting Keys** .......................................... 103
- Setting Preferences for Keying .......................... 105
- Methods for Setting and Removing Keys ................. 106
- Setting Keys at Multiple Frames ....................... 108
- Keying Parameters in the Keying Panel ................ 109
- Choosing Which Parameters to Display in the Keying Panel 111
- Changing Parameter Values in the Keying Panel ........ 113
- Setting Keys in the Keying Panel ....................... 114
- Muting Animation in the Keying Panel ................ 116
- Keying Parameters in Property Editors ................. 117
  - Keying All or Marked Parameters ..................... 117
  - Keying Specific Parameters .......................... 118
  - Moving between Keys in a Property Editor .......... 119
  - Removing Keys in a Property Editor ................. 119
- Setting Keys Automatically ............................ 121
- Setting Up for Autokeying ............................. 121
- Autokeying ................................................ 122
- Moving Between Keys ................................... 123

## Chapter 5: Editing Keys in the Timeline

- **Viewing Keys in the Timeline** ...................... 126
- **Editing Keys in the Timeline** ...................... 126
- View Audio Waveforms in the Timeline ................ 128
- Moving Keys in the Timeline ........................... 129
- Scaling Keys in the Timeline ........................... 131
- Cutting, Copying, and Pasting Keys in the Timeline .... 133
- Cutting Keys ............................................. 133
- Copying Keys ............................................. 133
- Pasting Keys ............................................. 134
- Merging or Replacing Existing Keys When Pasting .... 135
- Keying the Current Values at a Different Frame ....... 135
- Deleting Keys in the Timeline ........................... 137
- Setting Preferences for Keys in the Timeline .......... 138

## Chapter 6: Animating Transformations

- **Tools for Animating Transformations** ............... 140
- Animating Local versus Global Transformations ....... 142
  - Local Transformations ................................ 142
  - Global Transformations ................................ 142
- Manipulation Modes versus Transformation Values ...... 143
- Marking Transformation Parameters ................... 144
  - Marking Specific Local Transformation Axes .......... 144
  - Marking Global Transformation Parameters ........... 145
Contents

Keying Transformations .................................................. 146
Keying Transformations Quickly ................................. 147
Removing Transformations ........................................ 147
Animating Transformations in Hierarchies .................. 148
Changing the Animation Inheritance ..................... 148
Creating Transform Groups (Parenting with Null Objects) ....... 150
Animating the Transformation of Object Centers .......... 152
Animating Rotations .................................................. 153
Euler versus Quaternion Rotation ............................... 153
Converting between Euler and Quaternion Rotations ........ 154
Keying Quaternion Rotations ........................................ 155
Making Rotations Continuous ................................. 156
Visual Cues for Animated Transformations ............. 157

Chapter 7

Editing Function Curves ............................................. 159
Working in the Animation Editor ............................. 160
Opening the Animation Editor ................................. 160
Selecting the Animation Editing Tool ..................... 161
Using the Animation Tree ........................................ 162
Setting Preferences for the Animation Tree ........... 163
Using the Animation Editor’s Timeline .................. 164
Working in the Fcurve Editor .................................... 166
Setting Fcurve Editor Preferences ......................... 166
Commands and Tools for Editing Fcurves ................ 166
Undoing and Redoing Fcurve Modifications .............. 168
Opening the Fcurve Editor with Specific Fcurves Selected .... 168
Selecting Objects with Fcurves .............................. 169
Getting Fcurve Information ...................................... 169
Working with High-Density (Raw) Fcurves .............. 169
Setting Up the Fcurve Graph ................................... 171
Setting the Time Format (X Axis) ......................... 171
Setting the Graph’s Grid and Rulers ..................... 171
Viewing Function Curves .......................................... 173
Controlling Which Fcurves Are Displayed ............. 174
Locking on to an Element .......................................... 176
Hiding and Displaying Fcurves ............................... 177
Hiding and Displaying Fcurve Information ............ 177
Viewing Fcurves of Different Values (Normalizing) ....... 178
Displaying an Overview Region (Metacurve) ........... 180
Viewing Corresponding Time with Action Sources .......... 180
Zooming, Panning, and Framing in the Graph ............ 181
Zooming ................................................................. 181
Panning ......................................................................... 182
Panning and Zooming ............................................... 182
Framing ....................................................................... 183
Contents

Creating Function Curve Cycles .......................................................... 223
Creating a Basic Cycle ........................................................................ 223
Creating a Relative Cycle (Offset) ...................................................... 224
Deleting Fcurve Cycles ...................................................................... 224
Freezing Fcurve Cycles ...................................................................... 225
Editing Quaternion Function Curves .................................................. 226
Shaping High-Density Fcurves ............................................................ 229
Selecting an HLE Editing Operation .................................................. 231
Processing Function Curves ............................................................... 233
Reducing Noise (Smoothing) ............................................................... 233
Decreasing the Number of Keyframes (Fitting) ................................. 234
Changing the Number of Keyframes (Resampling) ............................ 235
Deactivating (Muting) Function Curves .............................................. 236
Converting Between Raw and Standard Fcurves ............................... 237
Applying Preset Function Curves ...................................................... 238
Saving and Loading Function Curves ................................................. 239
Saving Fcurves .................................................................................. 239
Loading Fcurves ................................................................................ 240

Chapter 8
Editing Animation in the Dopesheet ..................................................... 241
  Displaying the Dopesheet ................................................................. 242
  “Dopesheet” in the Timeline ............................................................. 243
  Setting Preferences for the Dopesheet .............................................. 243
Controlling What Is Displayed in the Dopesheet ............................... 244
Zooming, Panning, and Framing in the Dopesheet .............................. 245
Working with Tracks ........................................................................ 247
  The Summary Track ........................................................................ 247
  Expanding Tracks .......................................................................... 248
  Collapsing Tracks .......................................................................... 248
  Displaying or Hiding Tracks ............................................................ 249
  Aligning the Tracks with the Parameters in the Tree ....................... 249
  Marking Parameters Using Their Tracks ........................................ 250
Working with Keys ............................................................................ 251
Editing Keys Using Regions ............................................................... 252
  Drawing (Selecting) Regions of Frames .......................................... 252
  Rippling ......................................................................................... 253
  Deleting Keys from Regions .......................................................... 253
Moving Regions ................................................................................ 254
Scaling Regions ................................................................................ 255
Cutting, Copying, and Pasting Regions .............................................. 256
  Cutting or Copying a Region .......................................................... 256
  Pasting the Region ......................................................................... 257
Cutting, Copying, and Pasting Animation between Tracks ............... 259
Cutting, Copying, and Pasting Animation between Models and Hierarchies ... 259
Symmetry Constraints .................................................. 301
Constraints between Points ........................................... 303
  Constraining between Two Points ......................... 303
  Constraining an Object within Three Points .............. 304
  Constraining an Object within Multiple (N) Points ...... 305
Object to Cluster Constraints ..................................... 306
Surface and Curve Constraints .................................... 308
  Constraining to a Surface .................................. 308
  Constraining to a Curve ................................... 309
Bounding Volume and Bounding Plane Constraints ........ 310
Up-vector Constraints .............................................. 312
  About the Up-vector Constraint ......................... 312
  Constraining the Up Vector to an Object ............... 313
  Constraining the Up Vector without an Object ........ 315
  Controlling Flipping with Path Animation ............. 315
  Removing an Up-Vector Constraint .................... 315
Tangency Constraints ............................................... 316
Creating Offsets between Constrained and Constraining Objects 318
  Setting Offsets ........................................ 318
  Setting the Coupling Behavior between Objects ....... 319
  Setting Offsets with Compensation ................... 320
Blending Constraints .............................................. 321
  Blending Constraints with Each Other ................. 321
  Blending Constraints with Animation ................. 323
Activating and Deactivating Constraints .................. 324
  Removing (Relaxing) Constraints .................... 325

Chapter 11

Linking Parameters .................................................. 327
  Viewing Linked Parameter Information .................... 328
  Linking to a Single Driving Parameter ................. 330
  Setting Relative Values ................................. 332
  Editing Linked Parameter Fcurves ..................... 333
  Example: Linking the Opening Doors to a Missile .... 335
  Linking to Multiple Driving Parameters .............. 337
  Setting Relative Values for Multi-Linked Parameters . 338
  Editing the Falloff ..................................... 339
  Linking to an Object’s Orientation .................... 341
  Copying and Removing Link Animation ............... 342
  Tips & Tricks with Linked Parameters ............. 342

Chapter 12

Animating with Expressions ..................................... 345
  Viewing Expression Information .......................... 346
  Selecting Objects with Expressions .................... 348
  Jumping into Expressions ................................. 349
  Displaying the Expression Editor ..................... 352
Contents

Creating Expressions ................................................. 353
Creating an Expression Without Writing ......................... 353
Writing Expressions .................................................. 355
Tips for Writing Expressions ........................................ 356
Expression Syntax ..................................................... 356
Selecting or Changing the Target ................................... 356
Entering Object Names ............................................... 357
Entering Parameter Names ........................................... 358
Entering Functions and Constants ................................... 358
Operators .............................................................. 359
Using Another Text Editor ............................................ 360
Validating and Applying Expressions ............................... 361
Copying and Removing Expressions ................................. 361
Editing an Existing Expression ..................................... 362
Activating and Deactivating Expressions ......................... 364
Saving and Loading Expressions ................................... 365
Importing Expressions from SOFTIMAGE|3D ...................... 366
Expressions and Other Animation ................................... 367
Combining Expressions and Constraints ......................... 367
Combining Expressions and FCurves ............................... 367
Converting Expressions to Function Curves ....................... 368
Expression Examples .................................................. 369
Relational Expressions ............................................... 369
Conditional Expressions ............................................. 371
Oscillation Using Cosine ............................................. 372
Tips & Tricks with Expressions ..................................... 373
Multiple Expressions on the Same Parameter ..................... 373
Using Color Parameters with Expressions ....................... 373
Using Custom Parameters as Variables ........................... 374
Using Expressions on Object Components ....................... 374
Using Expressions on Preferred Angles ......................... 374

Chapter 13  

Device Drivers ....................................................... 375
Overview of Using Device Drivers .................................. 376
Managing the Device Drivers ....................................... 377
Adding and Removing Device Drivers .............................. 377
Activating and Deactivating Device Drivers ..................... 378
Setting Up the Device Driver’s Channels ......................... 379
Capturing Animation Information ................................. 381

Index ........................................................................ 383
Animation

This guide covers features about general and low-level animation in XSI:

- *Animating in SOFTIMAGE|XSI* on page 19
  An overview of all the ways in which you can animate in XSI, including some common tools used for animation.

- *Tasks for Any Type of Animation* on page 33
  Covers many animation tasks that are general to any type of animation, such as plotting, copying, or removing animation.

- *Playing Animation* on page 71
  Describes the different issues around playing back your animation, from setting up your frame rates and frame formats, to playback options and flipbooks.

- *Animating with Keys* on page 101
  Explains the many ways in which you can set keys, the most fundamental way of animating in XSI.

- *Editing Keys in the Timeline* on page 125
  Describes how to view and edit keys for animated objects in the timeline.

- *Animating Transformations* on page 139
  Covers the various issues surrounding animating transformations (scaling, rotation, and translation), from global and local transformations to keying transformation parameters, and much more in between.

- *Editing Function Curves* on page 159
  All about editing keyframed animation in the fcurve editor (sometimes known as the animation editor).

- *Editing Animation in the Dopesheet* on page 241
  All about editing keyframed animation in the dopesheet.
• Animating along Paths and Trajectories on page 265
  Describes the various ways in which you can animate objects on paths and trajectories.

• Animating with Constraints on page 283
  Covers the many different types of constraints available in XSI, such as direction, orientation, symmetry, and pose.

• Linking Parameters on page 327
  Explains how to link parameters together in an animation relationship.

• Animating with Expressions on page 345
  Describes how to create mathematical expressions to solve many animation problems.

• Device Drivers on page 375
  Describes how to set up device drivers for use with XSI in order to animate objects with the information captured from the attached device.
Chapter 1

**Animating in SOFTIMAGE®|XSI®**

To animate means to breathe life into something and life is always signified by change: growth, movement, dynamism. In SOFTIMAGE|XSI, everything can be changed. Everything in an XSI scene is represented by numeric values, and animation is the process of changing these values—position, color, or any other property. For example, you can make a cat leap on a chair, a light grow dim, a camera pan across a scene, a material change color, or fog evaporate.

The animation tools in XSI let you create animation quickly so you can spend the time you need to perfect it by editing the movement, changing the timing, and trying out different tools as often as you need to. XSI gives you the control and quick feedback you need to produce great animation.

**What Can You Animate in XSI?**

You can animate every scene element and most of their parameters—in effect, if a parameter exists on a property page, it can probably be animated. The only thing you can’t animate are global rendering options!

Animatable parameters fall into four categories:

- **Motion:** Probably the most common form of animation, this involves displacing an object from one point to another, or rotating it.

- **Geometry:** You can animate an object’s structure by changing values such as U and V subdivision, radius, length, or scale. You can also use surface deformations and skeletons to bend, twist, and contort your object.

- **Appearance:** Material, textures, visibility, and transparency are just some of the parameters controlling appearance that can be changed over time.

**How Do You Animate in XSI?**

In XSI, there are different levels at which you can animate. For example, you can animate at the lowest level by keyframing a specific parameter or you can animate at a high level by mixing together blocks of animation.
Between these extremes are many other methods such as creating secondary animation driven by an initial animation or by dynamically simulating natural phenomena.

Basically, if you want to make it move, XSI has the tools.

- As with other animation software, XSI’s most basic method of animation is keyframing. You set property values, or keys, at specific frames, called keyframes, and values for the frames in between are calculated by interpolation.

There are a multitude of other animation tools that give you control at the lowest level, such as constraints, expressions, custom and proxy parameters, and scripted operators.

See Chapter 2: Tasks for Any Type of Animation on page 33 and Low-level Animation on page 21 for more information.

- The character animation tools in XSI offer you control when creating and animating skeletons. You can animate them with forward or inverse kinematics, add an enveloping model, set up a rig, and fine-tune the skeleton’s movements in a myriad of ways to get just the right motion.

See the Character Animation guide for more information.
The animation mixer is a powerful editing tool that is similar to a video editing suite. Although you can edit, mix, and reuse animation at a high level, you always have control over the details.

Using action, shape, and audio clips, you can quickly and easily combine animation types, mix their weights, copy animation from one model to another, create animation sequences, or synchronize animation with audio.

See the Nonlinear Animation guide for more information.

Levels of Animation

One of the most important features of XSI is its layered approach to animation. Each layer represents a different level of control, from keying a single parameter’s value to copying animation from one character to another. This allows you to easily manage complex animation yet retain the ability to work at the most granular level.

It is important to understand the hierarchical nature of the control levels. Each level provides tools that are compatible with and make use of any levels below it. Tools can use other tools from any of the lower levels internally, but not from higher levels. For example, you can use the animation mixer to blend actions together in a transparent manner, whether the action contains function curves, constraints, or expressions.

Low-level Animation

Animating at a low level means getting down to the parameters of an object and animating their values. And in XSI, almost every parameter can be animated. While low-level animation techniques may be methodical in some ways, they are also powerful in the amount of control you can have.
Keyframing is the most common method of direct animation, but you can also use path animation, constraints, linked parameters, expressions, and scripted operators for “indirect” animation. These methods help speed up the animation process by creating relationships between parameters. As a simple example, you can set a constraint between one object and a multitude of objects, and then just animate that one object to have the multitude follow.

To see how many of the low-level animation features can be used together, see the Low-Level Animation lesson in the Tutorials guide.

For more information on low-level animation, see the rest of the chapters in this guide; for information about scripted operators, see Scripted Operators on page 151 in the Customization guide.

High-level Animation

Animating at a high level means that you are working with animation information in a way that is nonlinear (independent of the timeline) and non-destructive (all animation work does not destroy the original animation data).

High-level animation is done using the animation mixer, whether it be animation coming from function curves and expressions or from shapes that you’ve stored. Any type of animation that you generate can be stored and reused later, on the same or a different model. As well, you can mix different types of animation together at the same time and determine their weighting.

For more information on using the animation mixer, see the Nonlinear Animation guide for more information.

Order of Animation

If you’re animating an object using a number of different techniques, you need to understand how the different animation operations take precedence over each other. This is only an issue if you have more than one type of animation on the same parameter.

1. Function curves (keys), expressions (including linked parameters which are a special type of expression), and scripted operators are the base level.
2. Mixer connections take precedence over the base level.
3. Constraints (not including up-vectors and tangents, which aren’t “real” constraints) take precedence over mixer connections.

For example, if you have fcurves and an action source (in the mixer) driving a parameter, the mixer takes precedence over the fcurve.

There are two special cases involving animation on different parameters:

- Inverse kinematics (animation on a chain end effector’s position) overrides forward kinematics (animation on a chain bone’s rotation), no matter how the parameters are animated. However, you can blend inverse and forward kinematics on the same bone.
• Global transformations override local transformations, no matter how the parameters are animated.
Tools for Animation

A number of tool groups in the main window are designed specifically to help you accomplish your animation tasks. This section briefly introduces them.

The main place to get commands for animation is the Animate toolbar.

The Animate Toolbar

You'll find the Animate toolbar at the far left of the screen, as well as on the main menu bar at the top of the XSI window. It contains all commands related to animation.

Get commands
Create generic elements, including primitive objects, cameras, and lights.
These features are also available on Model, Render, and Simulate toolbars.

Create commands
Define expressions, animation paths, custom and proxy parameters, and skeletons.

Deform commands
Create and edit deformations, including shapes and envelopes.

Action commands
Save and apply actions to use in the animation mixer.

Tools commands
Plot animated movement, set up a device driver, and import and export animation.
General Tools for Animation

The Animation Panel
Located at the bottom-right of the XSI window, the Animation panel contains controls for quickly accessing a number of important animation options:

- **Animation menu** contains commonly used animation commands that set keys, display the animation editor, copy, paste, remove animation, and more.
- **Autokey button** automatically sets a key after each parameter edit.
- **Marked Parameter icon** opens the marked parameter list.
- **Keyframe icon** adds a key at the current frame on the timeline.
- **Marked Parameter list box** displays the currently marked parameter.
- **Arrow buttons** move between previous and next keyframes.

- The **Animation menu** contains commands used for many general animation operations such as setting and removing keys, moving between keyframes, copying and removing animation, scaling and offsetting animation, setting expressions and scripted operators, linking parameters, unmarking parameters, and opening the expression and animation editors.
- The **keyframe icon** lets you add a key at the current frame, while the **autokey button** automatically sets a key every time you change a parameter's value. You can also move between keyframes using the arrow buttons below the auto button.

For information on keyframing, see *Animating with Keys* on page 101.

- The **marked parameter** icon lets you open the marked parameter list when you want to mark parameters for certain animation operations. The current parameter is displayed in the text box below the list. You can also lock and clear marked parameters using the appropriate buttons.

For more information on marking, see *Marking Parameters for Animation* on page 35.

The Transform Panel
Transformations (scaling, rotation, and translation) are probably the most commonly used tools for animation. Located in the main command panel on the right of the XSI window, the Transform panel (as shown on the left) is where you decide how to scale, rotate, or translate any object in XSI.

For some important information about animating transformations, see *Animating Transformations* on page 139.

For information on transformations in general, see *Transformations* on page 27 in the Transformations guide.
The Animation Icon

You will see this in a lot of places in XSI, such as the explorer or any property editor: the little green icon beside any parameter indicating that it can be animated. This is called the animation icon and it lets you do a number of things.

- Click it to set or remove a key for that parameter at the current frame.
- Right-click it to use commands specific to the parameter, as well as to access animation tools such as the animation or expression editors.

Depending on the status of the parameter’s animation, the appearance of the icon changes:

- Default green and small: the parameter is animatable, but not currently animated.
- Red with a curve: the parameter is keyed at the current frame.
- Green with a curve: the parameter is animated but the current frame is not a keyframe.
- Yellow with a curve: the animated parameter’s value has been changed but not yet set as a keyframe. If you change the current frame without keyframing the parameter’s new value, the parameter reverts to its interpolated value.
- White with a curve: the animation for the parameter has been deactivated (muted) in the keying panel (see Keying Parameters in the Keying Panel on page 109).
- With a letter C and a connection: the parameter is controlled by a constraint (see Animating with Constraints on page 283).
- With a curve and a letter L: the parameter is a linked parameter (see Linking Parameters on page 327).
- With a letter S: the parameters is driven by a scripted operator (see Scripted Operators on page 151 in the Customization guide).
If the animation icon is cyan with a black arrow in it, it means that the parameter is controlled by an override. Overrides temporarily replace the current value of the parameter.

Right-click the animation icon and choose Inspect Override to open the override’s property editor. For more information on overrides, see Overriding Properties on page 112 in the Scene Elements guide.

These icons indicate expressions of different types (see Animating with Expressions on page 345):

- With a letter C: the parameter has an expression that is a constant value.
- With an = sign: the parameter has an expression that makes the parameter equal to another one (links them).
- With an arrow: the parameter has any type of expression on it other than a constant value or an “equal” link.

These icons indicate parameters driven by action clips in the animation mixer (see Actions on page 89 in the Nonlinear Animation guide):

- Green or red with a plug: the parameter is driven by an action clip.
- Red or green with a plug and a plus sign: the parameter is driven by an action clip as well as a function curve, expression, constraint, or scripted operator at the same time.

Playing the Animation

There are different ways of playing back animation in the viewports, but the most common ways are to use the timeline and the Playback panel.

You can also preview your animation in a render region prior to the final render. You can get real-time playback of animation in the render region by using its caching capabilities. See Capturing Animation in a Viewport on page 91.

The Timeline, Time Range, and Playback Controls

The timeline displays the current playback position of the animation and lets you manually move between different frames. The current frame is indicated by the position of the playback cursor (the vertical red bar). The time range (not displayed by default) lets you work with just a range of frames within the scene. The playback controls on the Playback panel below the timeline work in conjunction with the timeline, allowing you to play animation, as well as set up numerous options for playback.

For more information, see Playing the Animation on page 77.
Chapter 1

The SOFTIMAGE|XSI Flipbook

When you want to preview an animation in real time, you can capture animation sequences in a viewport and then play them back in the standalone flipbook.

For information, see Previewing Animation in a Flipbook on page 94.

Getting Animation Information

If your scene has any type of animation in it, you can find out about it by choosing Edit > Info Scene from the main command panel (or press Ctrl+Enter).

- On the Objects page, you can find the number of Chain Elements in the scene. The total number includes chain roots, bones, and effectors.

- On the Animation page, information about the following elements will show up, only if they are present in the scene:
  - Constraints
  - Mixers
  - Action (animation) sources, items, and keys
  - Shape sources
  - Audio sources

Setting Animation Preferences

When you set up XSI, there are a number of preferences you can set to customize it to suit your workflow. These options let you set the default style, interpolation, and slope modes for function curves when they are created (that is, when you are setting keys). There are also options for setting the default autokey behavior and keying in property editors.

To set the animation preferences

1. Choose File > Preferences from the main menu.

2. In the explorer that appears, select Animation or click its icon to open the Animation Preferences property editor.
Selecting Types of Animated Objects

When you have a crowded scene, it’s sometimes difficult to select only the objects you want. To make selecting animated objects easier, you can choose from many types of filters on the Filters menu on the Select panel. For example, you can choose a filter so that only chain elements (which includes a chain’s roots, bones, joints, and effectors) can be selected, or choose a filter so that only objects with expressions can be selected.

To use the selection filters

1. In the Select panel of the main command panel, click the little arrow button to open the Filters menu.

2. Select the type of filter according to the animation element you want to select:
   - Objects with Marking Sets—see Marking Parameters for Animation on page 35 for more information.
   - Objects with Fcurve Animation—see Chapter 7: Editing Function Curves on page 159 for more information.
   - Objects with Expressions—see Chapter 12: Animating with Expressions on page 345 for more information.
   - Objects with Constraint or Constraints themselves—see Chapter 10: Animating with Constraints on page 283 for more information.
   - Objects with Mixer Animation (animated by actions)—see Actions on page 89 in the Nonlinear Animation guide for more information.
   - Objects with Shapes—see Animating Shapes on page 10 in the Shape Animation guide for more information.
   - Chain Elements, Bones, Effectors, or Joints—see the Character Animation guide for more information.

Tools for Animation

See the online help in this property editor for more information, as well as Editing Function Curves on page 159 and Animating with Keys on page 101.

To set preferences for editing function curves after you’ve created them, see the Fcurve Editor Preferences property editor (choose File > Preferences and then select Editors > Fcurve Editor).
Finding Animation Elements

The typical viewing tools in XSI can be optimized to view animation elements and information about them: the explorer, the schematic view, and the spreadsheet.

In the Explorer

The explorer is a useful tool for viewing all elements in a scene. It presents the elements and their properties in a tree-like structure that you can expand or collapse to display as much detail as you like.

As with all multi-purpose tools, there are a few tricks for filtering out specific information. The explorer has a number of commands in the Show menu for filtering out animation information, such as Animated parameters and Animatable parameters. You need to select these filters if you want to see any animated parameters for an object.

You can tell if a parameter is animatable if it has an animation icon beside it, and if the parameter is animated, the animation icon has some animation symbol in it (see The Animation Icon on page 26).

Choose Show > Parameter Values to display the values beside each parameter which you can then edit using virtual sliders. See Entering Values Using Virtual Sliders on page 57 in the Interface and Tools guide.

In the filter menu beside Show, you can also select All+Animatable Parameters to further filter for animation elements. If you’re using the animation mixer, choose Mixers Only in this menu to have only the mixer node for each model appear.

There are many ways in which an object and its animation properties are displayed in the explorer, depending on the type of animation that you are doing. The most common way of animating is transforming. When you transform an object (scale, rotate, or translate), you will find the information in the object’s Kinematics node (kinematics meaning “movement” in general, not “inverse or forward kinematics”).

For information on other nodes that are specific to animation, see these pages:

- Under the Kinematics node, you will find the Local and Global Transform nodes, which contain the scaling (Scl), rotation (Ori > Euler), and translation (Pos) parameters (see Animating Transformations on page 139).

- Under the Kinematics node, you can also find Constraints, which displays any constraints set for the object (see Animating with Constraints on page 283).

- For information on skeleton/envelope nodes, see the Character Animation guide.
Finding Animation Elements

- For information on action nodes, see Actions on page 89 in the Nonlinear Animation guide.
- For information on shape key nodes, see Animating Shapes on page 10 in the Shape Animation guide.

For more information in general about using the explorer, see The Explorer on page 83 of the Interface and Tools guide.

In the Schematic View

The schematic view shows the relations between the objects in your scene, such as the hierarchy of objects and how objects constrain one another.

Links between nodes appear by default. To toggle links on and off, choose Show > Constraint Links. Double-clicking a link displays its property editor.

You can see which objects are animated because they have the letter A on their nodes for animated (with keys), C for constraints, E for expression, S for shape animation, and so on.

Constraint links appear in green and display an abbreviated label indicating the type of constraint in effect, such as Dir for a direction constraint.

Expression links appear in green and are identified by the label EXPR. Expression information is only displayed for local and global transformations.

For more information on the Schematic view, see The Schematic View on page 95 of the Interface and Tools guide.
In the Spreadsheet

The spreadsheet shows information about the objects in your scene. For animation information, you can use the predefined queries available for fcubes, expressions, constraints, and chains/skeletons. The following shows the kind of information you can get for the skeletons in a scene.

For more information on the spreadsheet, see The Spreadsheet on page 106 of the Interface and Tools guide.
Chapter 2  **Tasks for Any Type of Animation**

This chapter covers a number of general issues and tasks that can apply to any animation technique or tool in XSI.

• Marking parameters is a way of telling XSI what you intend to animate so that it keeps just the right amount of information. You can mark parameters for many animation purposes, including setting keys, copying and removing animation, or storing animation in an action.  
  
  See *Marking Parameters for Animation* on page 35 for more information.

• Animation ghosting, also known as onion-skinning, lets you display a series of snapshots of animated objects at different frames or keyframes. This lets you easily visualize the motion of an object, which can help you improve its timing and flow.  
  
  See *Ghosting Animated Objects* on page 42 for more information.

• You can copy and paste animation from objects or parameters using different tools.  
  
  See *Copying Animation* on page 52 for more information.
- You can change the length, offset, or retime any type of animation for a whole scene, for selected objects, or for just marked parameters.

See *Scaling and Offsetting (Retiming) Animation* on page 55 for more information.

- Plotting animation “bakes” together any kind of animation information on a parameter into a single function curve. This is useful when you want to bring different types of animation together into one source or if you’re exporting animation.

See *Plotting Animation* on page 59 for more information.

- Muting animation helps you isolate parts of an animation so that you can focus on specific areas at a time, making sure that you get each part of the animation working properly before you bring it all together.

See *Muting (Deactivating) Animation* on page 65 for more information.

- Building animation relationships involving constraints, expressions, and scripted operators can often lead to evaluation cycles. XSI implements internal cycle checking to prevent cycles from occurring in a scene. As well, there are tools to help you run cycle checks to prevent problems.

See *Checking for Animation Dependency Cycles* on page 66 for more information.

- If you have some animation you need to get rid of, you can remove different types of animation from objects or parameters using different tools.

See *Removing Animation* on page 69 for more information.
Marking Parameters for Animation

Marking parameters is a way of selecting which parameters you want to use for a specific animation task. For example, you mark parameters so that only their animation is copied or removed, or you mark the parameters you want to store in an action to be used in the animation mixer. You can also mark parameters to be used when scaling, offsetting, or plotting an animation, or with linked parameters or scripted operators.

Marking parameters is also one of the most common and useful tools for setting keys. By marking only the parameters you need, you can keep the animation information small. For more information on keying, see Chapter 4: Animating with Keys on page 101.

By default, when you activate any of the transformation tools (scale, rotate, or translate) from the Transform panel or using their hot keys, their corresponding local transformation parameters are automatically marked. For more information on this, see Marking Transformation Parameters on page 144.

As well, there are a number of issues related to marking parameters when saving actions that are to be used in the animation mixer. For more information on this, see Choosing What to Store in an Action on page 100 in the Nonlinear Animation guide.

You can mark parameters for selected objects only. As well, the same parameters remain marked for an object even if the selection changes. For example, if you select a sphere and mark its scaling parameters, then select a cube, the scaling parameters are also marked on the cube. However, any marked parameters that do not apply from one object to the next are ignored.

You can also create marking sets to remember the parameters that were marked on an object—see Remembering Marked Parameters with Marking Sets on page 39.

To quickly change the values of marked parameters, use the “virtual sliders” in the explorer. To use this, press F4 and use the Shift or Ctrl keys while middle-clicking to do fine or coarse value scrubbing, respectively. For more information, see Enter Values Using Sliders on page 57 in the Interface and Tools guide.

Setting a Preference for Marked Parameters

Before you start marking parameters and setting keys, you need to make sure that the keying preference is set to marked parameters.

To set the keying preference

1. Choose File > Preferences from the main menu and select Animation, or open the Animation menu in the Animation panel at the bottom of the XSI window.

2. Select the Save Key Command Mapping > On marked parameters option.
This option means that you must first mark parameters before setting keys on them. This is the default method for working in XSI.

You can also select the On marked keyable parameters option to set keys only on keyable parameters that you have marked in the keying panel (see Keying Parameters in the Keying Panel on page 109 for more information).

3. Set keys on the marked parameters using the methods described in Methods for Setting and Removing Keys on page 106.

Marking Parameters

You can mark parameters from the marked parameter list, the explorer, any property editor, or the keying panel. You can also mark parameters via their tracks in the dopesheet—see Marking Parameters Using Their Tracks on page 250.

Press Shift+click to mark multiple parameters or folders and Ctrl+click to toggle their marking.

To mark parameters from the marked parameter list

1. Select the element whose parameters you want to mark.
2. Click the arrow icon on the Animation panel that opens the marked parameter list.
3. In the list, expand a property node until you display the parameters you want to mark. Animatable parameters are indicated by an animation icon (green box) beside their names.
4. Click the parameter name to mark it, or Shift+click to mark multiple parameters. The parameter name is highlighted in yellow, indicating that it has been marked. Click the name again to unmark it.

Click a parent property or folder name to mark all its parameters in a hierarchy. For example, you can mark the whole Local Transform node to mark all transformations (scaling, rotation, and translation) for all three axes (this works only in the marked parameter list). The parent/folder is highlighted in yellow and its children in pale yellow.
You can also enter a name of a parameter in the marked parameter text box to mark it. For example, enter \texttt{kine.local.pos.posy} to mark the selected object’s local Y translation.

5. Click anywhere outside the marked parameter list to close it.

**To mark parameters in the keying panel**

- See Keying Parameters in the Keying Panel on page 109 for more information.

**To mark parameters in a property editor**

1. Select the element whose parameters you want to mark.
2. Open a property editor with the parameters you want to mark.
3. Click the parameter name to mark it or Shift+click to mark multiple parameters.

Animatable parameters are indicated by an animation icon (green box) beside their names. You can also mark non-animatable parameters.

The parameter name is highlighted in yellow, indicating that it has been marked.

**To mark parameters in the explorer**

1. Expand an element’s property tree until you display the parameters you want to mark.

Animatable parameters are indicated by an animation icon (green box) beside their names. You can also mark non-animatable parameters.
2. Click the parameter name to mark it or Shift+click to mark multiple parameters.

The parameter name is highlighted in yellow, indicating that it has been marked. Click the name again to unmark it.

Click a folder name to mark it and the parameters it contains. The parent/folder is highlighted in yellow and its children in pale yellow.

The parameters must be directly below the current folder for them to be marked; otherwise, you simply select the folder. For example, you can mark the Pos folder to mark all three axes, but you can't mark the Local Transform node to mark all transformations.

Marking Custom Parameters and Sets

You can mark a whole custom parameter set or just individual parameters within the set. For example, you can store actions for individual custom parameters in a set or for the whole set itself.

- To mark a whole custom parameter set, select the object for which the custom parameter set exists and open the marked parameter list. You need to select the object because marking is relative to selection.

Then click the name of the custom parameter set—you'll see all its parameters highlighted in yellow too. You can't mark the set in the explorer because it supports only “leaf” marking.

- To mark the individual custom parameters, click them in the explorer of the Marked Parameter list as you would any other parameter.

Locking Marked Parameters

To ensure that your marked parameters don’t get deleted or overwritten by other markings, click the Lock Marked Parameter icon on the Animation panel. This locks all marked parameters for all objects in the scene.

Locking is especially important to do if you’re using transformations (scaling, rotation, or translation). Activating any of the tools from the Transform panel automatically marks the appropriate scale, rotate, or translate parameters, but also has the effect of removing any existing marking from other parameters.

Locked parameters cannot be cleared—make sure to unlock them first (click the Lock icon again) if you want to clear them.
Marking Parameters for Animation

Clearing (Unmarking) Marked Parameters

To clear all marked parameters

- Click the Clear Marked Parameter (Clr) icon.
  
  or
  
- Choose Animation > Unmark All Parameters in the Animation panel.
  
  or
  
- In the explorer, right-click a parameter’s animation icon and choose Unmark All Parameters.

If you hold down the supra key while using a transform tool, your existing marked parameters will only be temporarily unmarked, even if you haven’t locked them. After you release the supra key, they will be marked again.

Remembering Marked Parameters with Marking Sets

Marking sets are lists that are used to remember which parameters are marked on an object (like an absolute path to the marked parameters). This makes it easy to work with hierarchies because each object within that structure can have its own marking set, such as a set or marked rotation parameters for bones, and a set of marked translation parameters for effectors, etc.

Marking sets are actually custom parameter sets with proxies of the marked parameters in them. As a result, they are saved as properties of each object, as are standard custom parameter sets.

You can also key all keyable parameters in the keying panel, making the keying panel like a marking set for the object. See Keying Parameters in the Keying Panel on page 109 for more information.

Creating Marking Sets

When you create a marking set, you store the current marked parameters in a set for each selected object. If a marking set already exists for the object, it is replaced by the new set (that is, you can have only one marking set per object at a time).

While you can rename a marking set like anything else in the explorer, this makes it become “inoperative.” However, you can create several marking sets on an object and use a script to switch names.

To create marking sets

1. Select an animatable object and mark the desired parameters as described in Marking Parameters on page 36.

2. Choose Animation > Parameters > Create Marking Set from the Animation panel at the bottom of the XSI window or press Ctrl+Shift+m.
   
   A marking set is created as a property of that object.

3. Repeat these steps for each object for which you want a marking set.
Chapter 2 • Tasks for Any Type of Animation

To use marking sets

When you’re animating, select one or more objects with marking sets and do any of the following:

- Choose Animation > Parameters > Activate Marking Set (or press Ctrl+m) to activate the marking set, then save a key or an action, etc.
- Choose Animation > Set Key Using Marking Set. This directly sets a key at the current frame using the marking set without first activating it or changing the currently marked parameters.
- Choose Animation > Set Branch Key Using Marking Set (or press Alt+k). This visits all nodes in a branch selection and keys according to the marking sets it finds. This is useful for working with characters.

To add marked parameters to a set

- Mark new parameters for an object and choose Animation > Parameters > Add to Marking Set.

To remove marked parameters from a set

- In the marking set, select the parameters you want to remove and choose Animation > Parameters > Remove from Marking Set.

To delete a marking set

- Select one or more objects with marking sets and choose Animation > Parameters > Delete Marking Set.

To select only objects that have marking sets, select the Obj w Marking Set filter from the Filters menu (click the arrow button in the Select panel to open the list).

- Choose Animation > Parameters > Activate Marking Set (or press Ctrl+m) to activate the marking set, then save a key or an action, etc.
- Choose Animation > Set Key Using Marking Set. This directly sets a key at the current frame using the marking set without first activating it or changing the currently marked parameters.
- Choose Animation > Set Branch Key Using Marking Set (or press Alt+k). This visits all nodes in a branch selection and keys according to the marking sets it finds. This is useful for working with characters.
Locking Animation for Objects and Parameters

When you create a scene or model, there may be certain parts of it that you don’t want to change or that you don’t want anyone else to change. To help reduce the introduction of error, you can use locks to control the type of modifications allowed to parts of a scene. For example, if you’re a technical director, you can lock the animation of certain parameters for a character so that they’re available, but can’t be keyed.

For all information on how to lock animation for objects and parameters, see Locking and Tagging Scene Elements on page 99 in the Data Management guide.

Locking Animation

When a scene element is locked at the animation level, you cannot set keys for it nor can you edit its fcurves in the animation editor.

In the fcurve editor, locked fcurves are drawn with dashed lines, as shown here.

You cannot select any locked fcurves, therefore you cannot select any keys on them and edit them. If you want to copy keys from fcurves, you can use the dopesheet.

In the dopesheet:

- Locked tracks have a black lock icon displayed at their left end. If a collapsed track contains all locked tracks, it has a black lock icon. If it contains a mix of locked and unlocked tracks, the lock icon is a mixture of black and green.
- You can select and/or create regions on locked fcurves (tracks), mostly for copying and pasting keys from them, but you can’t edit the keys.

If you try to edit the keys, an error message appears in the script window telling you that this cannot be done because the underlying fcurve is locked.

For more information on fcurves and the dopesheet, see Chapter 7: Editing Function Curves on page 159 and Chapter 8: Editing Animation in the Dopesheet on page 241.
Chapter 2 • Tasks for Any Type of Animation

Ghosting Animated Objects

Animation ghosting, also known as onion-skinning, is a viewing mode that lets you display a series of snapshots of animated objects at frames or keyframes behind and/or ahead of the current frame. This lets you easily visualize the motion of an object, which can help you improve its timing and flow.

Ghosting has its roots in cel animation, where animators use translucent sheets containing the previous and next keyframes as a reference while drawing. This gives the animator a feel for the motion of the object and serves as a guide when editing keyframes.

Ghosting works for any object that moves in 3D space, either by having its transformation parameters (scaling, rotation, and translation) animated or by having its geometry changed by shape animation or animated deformations (including envelopes, rigid body, cloth, and soft body). The transformation parameters can be animated in any way: by fcurves (keys), constraints, expressions, scripted operators, or an action clip in the animation mixer.

Ghost images themselves are not cached but are instead drawn for the animated object at each frame. As a result, the ghost images are updated automatically when you change the object’s animation. You cannot select or modify ghosts, and they cannot be rendered.

To see ghosting, you must activate it in a 3D view as well as activate it for the animated object. You can set ghosting per object, per layer, and per group.

You can also view ghosts for:

• An object that has overlapping clips in the animation mixer (mixer ghosting). This helps you blend the animation from two or more actions. See Ghosting Clips on page 39 in the Nonlinear Animation guide.
Ghosting Animated Objects

The blending between inverse and forward kinematics on the same chain (IK/FK blending). See Ghosting FK/IK Blending on page 147 in the Character Animation guide.

Activating Ghosting for Objects

You can set ghosting on a per-object basis. Remember that layer and group settings override individual object settings; for example, if ghosting is active for an object but not its layer, you won’t be able to see the object’s ghosts.

To activate the ghost display per object

1. In an explorer, click the Visibility icon for an object to open its Visibility property editor.
2. In the Animation area, select the Ghosting option.

If this option has a check mark but is unavailable (grayed out), it means that this object belongs to a layer or group whose ghosting setting is active (see next section, Activating Ghosting for Layers and Groups).

To toggle ghosting for selected objects

- Choose Display > Ghost/Unghost Selection from the main menu at the top of the XSI window.

This toggles animation ghosting display in the viewports for selected objects, overriding the ghost settings for the object, its layer, or its group.

To deactivate ghosting for all objects in a scene

- Choose Display > Unghost All Objects from the main menu.

This deactivates the animation ghosting display in the 3D views for all objects in the scene, overriding the ghost settings per object, layer, or group.

Activating Ghosting for Layers and Groups

In addition to setting ghosting per object, you can also set ghosting for all animated objects in a layer or group. Remember that layer and group settings override individual object settings.

To activate ghosting for animated objects in layers

- Do any of the following:
  - Click the KP/L tab at the bottom of the main command panel to open the Layer panel. Then choose View > Show Ghosting Visibility in the Layer panel. This activates ghosting for objects in the current layer.
  - In the layer control in the Layer panel, click in the Ghost cell for the layers on which you want to have ghosting.
To activate ghosting for animated members in groups

1. In an explorer, click the appropriate group icon to open its Group property editor.
2. Select Ghost members as the Animation Ghosting option. This activates ghosting for only the objects in that group.

To determine ghosting behavior for members in a layer/group

1. Click the layer/group icon in the explorer to open its Layer/Group property editor.
2. Select an option in the Animation Ghosting list for the members:
   - Do not ghost members prevents ghosting for the animated objects. This overrides the Animation Ghosting settings in the members’ individual Visibility properties.
   - Ghost members displays ghosts for the animated objects. This overrides the Animation Ghosting settings in the members’ individual Visibility properties.
   - No effect on members: members are ghosted or not according to the Animation Ghosting settings in their individual Visibility properties. This does not override the members’ Visibility properties.

Activating Ghosting in 3D Views

To see the ghost shapes for animated objects, you must activate ghosting for any 3D view in a viewport, object view, or the XSI explorer. Ghosting works in conjunction with any of the display types for the 3D views, such as Wireframe, Shaded, and Textured. Ghost shapes, however, are always drawn in wireframe if the object’s geometry is being used as the ghost shape—see Selecting the Ghosting Display Type on page 49 for other display options.

To activate ghosting display in a 3D view

1. Do one of the following:
   - Choose the Animation Ghosting command in the display types menu.
   - Click the appropriate layer icon in the explorer to open its Layer property editor, then select Members are ghosted as the Animation Ghosting option. This activates ghosting for only the objects in that layer.
Ghosting Animated Objects

- Choose Display Options in the display types menu to open the Camera Display property editor. Then click the Ghosting tab and select Enable Animation Ghosting.

2. Set up the ghost display options in the Camera Display property editor as described in the next section, Controlling the Ghost Display.

If you activate ghosting in a 3D view but still don’t see the ghosts, check that ghosting is active for the layer to which the objects belong—see Activating Ghosting for Layers and Groups on page 43.

To activate ghosting for all 3D views in viewports

1. Choose Display > Display Options (All Cameras) from the main menu to open the Camera Display property editor for all viewports.

2. Click the Ghosting tab and select Enable Animation Ghosting.

Controlling the Ghost Display

You can view ghosting in any 3D view. Although you can use ghosting with any display type (Shaded, Textured, Hidden Line, etc.), the ghost’s geometry is always drawn as wireframes. You can choose to draw the ghosts on top of objects, as well as have the ghosts fade away (become more transparent) the farther they get (in time) from the current frame.

By default, the object’s geometry is used as the ghost shape, but you can change the display type per object—see Selecting the Ghosting Display Type on page 49 for more information.

You can have ghosts on any interval of frames before and/or after the current frame, keyframes set on an object’s transformation parameters, and on static frames. As well, you can set ghosts for any combination of frames, keyframes, and static frames. For example, you can display static frames in one color and then display the interpolated frames between them in a different color.

To control the ghost display in a 3D view

1. Choose Display Options in the display types menu to open the Camera Display property editor.

2. Click the Ghosting tab and select Enable Animation Ghosting.

3. Set the Animation Ghosting parameters to set ghosts on:

   - Any combination of frames (see Drawing Ghosts on Frames on page 46).
   - Keyframes (see Drawing Ghosts on Keyframes on page 47).
   - Static frames (see Drawing Static Ghosts on Specific Frames on page 48).
Chapter 2 • Tasks for Any Type of Animation

Drawing Ghosts on Frames

You can display ghosts on any interval of frames before and/or after the current frame.

To set ghosts on frames

- Set the parameters in the Frame Options section:
  - Set the Frames Before value to determine the number of frames on which to display ghosts before the current frame. These are frames which have already played (have happened before the current frame).
    Set the color of these ghosts using the Before Color sliders.

  - Set the Alpha value of the color to make the ghosts more transparent.

    - Set the Frames After value to determine the number of frames on which to display ghosts after the current frame. These are the upcoming frames (will happen after the current frame).
    Set the color of these ghosts using the After Color sliders.

  The Frames Before and Frames After sliders only go up to 10, but you can enter a higher value in the text box.

Displays ghosts on top of objects in the view.

Ghosts fade out (become more transparent) as they get farther in time from the current frame.

Number of subdivisions that affects the smoothness of the motion trail display.

Size of the velocity vector when set as the ghost display type.

Displays ghosts only on keyframes set on local or global transformation parameters.

Sets the corresponding color of the ghosts on frames before and/or after the current frame.

Sets the corresponding color of the ghosts on keyframes before and/or after the current frame.

If you’re using FK/IK ghosting for FK/IK blending on a chain, you set the FK and IK ghost colors here.

Displays ghosts on any interval of frames—see below.

Sets the corresponding color of the ghosts on frames before and/or after the current frame.

Displays ghosts on top of objects in the view.

Ghosts fade out (become more transparent) as they get farther in time from the current frame.

Number of subdivisions that affects the smoothness of the motion trail display.

Size of the velocity vector when set as the ghost display type.

Displays ghosts on any interval of frames—see below.

Sets the corresponding color of the ghosts on frames before and/or after the current frame.

Displays ghosts only on keyframes set on local or global transformation parameters.

Sets the corresponding color of the ghosts on keyframes before and/or after the current frame.

If you’re using FK/IK ghosting for FK/IK blending on a chain, you set the FK and IK ghost colors here.

Draws ghosts only on keyframes set on local or global transformation parameters.

Sets the corresponding color of the ghosts on frames before and/or after the current frame.

Displays ghosts on any interval of frames—see below.

Sets the corresponding color of the ghosts on frames before and/or after the current frame.

Displays ghosts only on keyframes set on local or global transformation parameters.

Sets the corresponding color of the ghosts on keyframes before and/or after the current frame.

If you’re using FK/IK ghosting for FK/IK blending on a chain, you set the FK and IK ghost colors here.

**Drawing Ghosts on Frames**

You can display ghosts on any interval of frames before and/or after the current frame.

To set ghosts on frames

- Set the parameters in the Frame Options section:
  - Set the Frames Before value to determine the number of frames on which to display ghosts before the current frame. These are frames which have already played (have happened before the current frame).
    Set the color of these ghosts using the Before Color sliders.

    - Set the Alpha value of the color to make the ghosts more transparent.

    - Set the Frames After value to determine the number of frames on which to display ghosts after the current frame. These are the upcoming frames (will happen after the current frame).
    Set the color of these ghosts using the After Color sliders.

    The Frames Before and Frames After sliders only go up to 10, but you can enter a higher value in the text box.

Displays ghosts on top of objects in the view.

Ghosts fade out (become more transparent) as they get farther in time from the current frame.

Number of subdivisions that affects the smoothness of the motion trail display.

Size of the velocity vector when set as the ghost display type.

Displays ghosts on any interval of frames—see below.

Sets the corresponding color of the ghosts on frames before and/or after the current frame.

Displays ghosts only on keyframes set on local or global transformation parameters.

Sets the corresponding color of the ghosts on keyframes before and/or after the current frame.

If you’re using FK/IK ghosting for FK/IK blending on a chain, you set the FK and IK ghost colors here.

**Drawing Ghosts on Frames**

You can display ghosts on any interval of frames before and/or after the current frame.

To set ghosts on frames

- Set the parameters in the Frame Options section:
  - Set the Frames Before value to determine the number of frames on which to display ghosts before the current frame. These are frames which have already played (have happened before the current frame).
    Set the color of these ghosts using the Before Color sliders.

  - Set the Frames After value to determine the number of frames on which to display ghosts after the current frame. These are the upcoming frames (will happen after the current frame).
    Set the color of these ghosts using the After Color sliders.

  The Frames Before and Frames After sliders only go up to 10, but you can enter a higher value in the text box.
- Set the Frame Step value to determine the number of frames to display between ghosts. For example, a value of 2 displays a ghost on every other frame.

  Ghosts with Frames After set to 10 and a Frame Step of 3. Ghosts are faded as they get farther in time from the current frame.

  * Select Anchor Frames to keep the ghosts stationary with respect to the animated object, which makes it easier to evaluate the object's motion.

You can notice the effect of this option when the Frame Step value is high. The position of the ghosts change from frame to frame because they are relative to the current frame. This option positions the ghosts at absolute frames.

**Drawing Ghosts on Keyframes**

Drawing ghosts only on keyframes lets you quickly see where an object's keys are. If you add or remove a key with ghosting on, the ghosts are immediately updated to reflect the change.

Drawing ghosts on keyframes works only with keys that are set on an object's local or global transformation parameters.

**To set ghosts on keyframes**

  * Set the parameters in the Keyframe Options section:
    
    - Set the Keys Before value to determine the number of keyframes on which to display ghosts before the current frame. These are frames which have already played (have happened before the current frame).

    Set the color of these ghosts using the Before Color sliders.

    Set the Alpha value of the color to make the ghosts more transparent.

    - Set the Keys After value to determine the number of frames on which to display ghosts after the current frame. These are the upcoming frames (will happen after the current frame).
Set the color of these ghosts using the After Color sliders.

The Keys Before and Keys After sliders only go up to 10, but you can enter a higher value in the text box.

- Set the Key Step value to determine the number of keyframes to display between ghosts. For example, a value of 2 displays a ghost on every other keyframe.

Drawing Static Ghosts on Specific Frames

In addition to a sequence of frames or keyframes, you can also specify exactly which frames on which you want to display ghosts. This lets you see the state of the animation at any frame. Because these ghosts remain static as the animation is played back, they are useful placeholders. However, if you change the object’s animation, the ghosts are updated to reflect this.

To set ghosts on specific frames

1. In the Static Frames text box, enter the frame numbers (separated by commas) on which to display ghosts.

2. Set the color of the ghosts on static frames with the Static Color sliders.

You can also select the Draw Ghosts on Top option to have the static ghosts on top of the object’s geometry (see Drawing Ghosts on Frames on page 46).
Selecting the Ghosting Display Type

You can determine the appearance of the ghosted shapes for each animated object. By default, the object’s geometry is displayed as the ghost.

The display types use the current settings as defined in the Camera Display property editor (see Controlling the Ghost Display on page 45), including fading and whether ghosts are drawn on top of the object.

To select the ghost display type

1. In an explorer, click the Visibility icon for an object to open its Visibility property editor.

2. In the Animation area, select an option from the Ghost Type list: Object, Point, Pose, Velocity, or Trail.
For the Point, Pose, and Velocity display types, an icon is drawn at the current position in addition to one drawn for each ghost frame.

Ghosts displayed as **Objects**, which draws the object's geometry for each ghost.

Ghosts displayed as **Points**, which draws a point (small cross) for each ghost based on the center of the object's bounding box.

Ghosts displayed as **Poses**, which draws a transformation axis for each ghost based on the center of the object's bounding box.

Ghosts displayed as **Trail**, which draws a smooth curve connecting points created for each ghost. This displays the trajectory of the object.

Ghosts displayed as **Velocity**, which draws a velocity vector (direction arrow) for each ghost.
For the **Velocity** display type, you can set the size of the vector drawn with the **Velocity Scale Factor** in the Camera Display property editor (see *Controlling the Ghost Display* on page 45). The default value of 1 draws the vector in full size (Softimage units per second).

If you want to see the object’s velocity at only the current frame, you could set Frames After to 1 and the Frame Step to a large value like 1000.

### Setting the Trail’s Subdivisions

The curve created with the **Trail** display type is generated from the ghost points. To increase its accuracy, you can increase the number of subdivisions used to draw it.

**To set the trail’s subdivisions**

1. Select **Trail** as the Ghost Type in the object’s Visibility property editor.
2. In the Animation Ghosting area on the Ghosting page in the Camera Display property editor, set the number of **Trail Subdivisions**.

   The trail is drawn between the first and last ghost frames using the number of ghost frames and the Trail Subdivision value to sample the motion at regular intervals. A smooth curve is then drawn through the sample points. Higher settings make the trail more closely match the underlying motion, but take longer to compute.
Copy Animation

You can copy animation from selected objects or marked parameters using any of the numerous commands from the Animation menu in the Animation panel. As well, using the explorer or a property editor, you can easily copy and paste animation between individual parameters.

For keyframed animation, you can also use the animation editor to copy a whole function curve or just certain keyframes (key points) on it (see Cutting, Copying, and Pasting in the Graph on page 206), copy and paste keys in the timeline (see Cutting, Copying, and Pasting Keys in the Timeline on page 133), or use the dopesheet to copy and paste keys (see Cutting, Copying, and Pasting Regions on page 256).

Using the Animation Menu

To copy animation between objects

1. Select the object or mark the parameters from which you want to copy the animation.

2. From the Animation panel, choose Animation > Copy Animation and select any of these options in the menu:

   - Choose from Marked Parameters to copy all animation from the selected objects’ marked parameters to the clipboard.
   - Choose from Transforms to copy all scaling, rotation, and translation parameters. You can also choose to copy each of these types of transformations individually (X, Y, and Z parameters for each type) with their appropriate commands.
   - Choose from All Parameters, Any Animation Type to copy all animation from all parameters on selected objects to the clipboard.
   - Choose from All Parameters, Expressions to copy only expressions from all parameters on the selected object to the clipboard.
   - Choose from All Parameters, Fcurves to copy only fcurves from all parameters on selected objects to the clipboard.
Pasting Animation

To paste the animation

1. Select the object or mark the parameters to which you want to paste the animation.

2. Choose Animation > Paste Animation and either of these options:

   - Choose to All Parameters to paste the animation currently in the clipboard and apply it to all parameters of selected objects (based on parameter name matching).

   - Choose to Marked Parameters to paste the animation currently in the clipboard and apply it to the marked parameters.

For example, if the clipboard contains animation for all transformations, but only the rotation parameters are marked, only the rotation parameters are pasted.

Using the Explorer or a Property Editor

Another way in which you can copy animation is at the parameter level using the explorer or property editors. This can be animation from keys, linked parameters, expressions, or scripted operators. Animation can be duplicated between different parameters for the same object or for different objects. For example, you could copy the X rotation values from one object to the Y scaling of another object.

To copy animation in the explorer or a property editor

1. Open an explorer or a property editor. If necessary, choose to show All+Animatable Parameters from the explorer command bar.

2. Do any of the following:

   - Right-click the animated parameter’s animation icon and choose Copy Animation. Then right-click the target parameter’s animation icon and choose Paste Animation.

   - With keyframed animation, drag-and-drop the animated parameter’s animation icon to another parameter’s animation icon. This works only in a property editor.

If you do this for unanimated parameters or linked parameters, you create an “=” expression instead, meaning parameter A = parameter B (see Creating an Expression Without Writing on page 353).
Chapter 2 • Tasks for Any Type of Animation

or

- With keyframed animation, drag-and-drop the animated parameter’s name to another parameter. This works only in the explorer.

or

- In the explorer, drag-and-drop an entire animation folder from the animated object to another object at the same level (see the following example).

Example

You can copy an entire folder by dragging and dropping it from one object’s tree onto the other object’s folder of the same name.

1. Get two primitives, a sphere and a cube.
2. Animate the sphere’s translation.
3. In the explorer, expand the sphere’s Kinematics > Local Transform > Pos tree, and do the same for the cube.
4. In the sphere’s tree, click the name of the Pos folder and drag it to the Pos folder for the cube to copy over the translation animation for all three axes.

Pressing Shift while dragging and dropping moves the animation instead of copying it.

Copying Animation between Models and Hierarchies

Using the dopesheet, you can cut/copy and paste animation from one model to another, or from one hierarchy of objects to another within the same model. The objects in the hierarchy or model must have the same parameter names (or a subset or superset of them).

For information on how to do this, see Cutting, Copying, and Pasting Animation between Models and Hierarchies on page 259.

This is similar to copying actions from one model to another (see Copying Action Sources between Models on page 172 in the Nonlinear Animation guide). For example, you can paste a walk cycle animation from the Bob model to the Fred model as long as Fred has the same parameter names as Bob.

If the models have submodels, their fcurves won’t be copied and pasted.
Scaling and Offsetting (Retiming) Animation

If you find that your whole animation is a bit too long or too short, or you just want to set it off by a few frames, you can do so with the Sequence commands. They give you control over animation by offsetting or scaling (stretching or shrinking) the motion of all or selected objects, or even just the marked parameters of selected objects. You can offset or scale all animation sources, function curves, and even clips in the animation mixer.

You can use the dopesheet or select all keys in the timeline to scale and offset animation globally. Using the dopesheet may be useful if you encounter some problems with animation not scaling properly with the Sequence commands.

For information, see Chapter 8: Editing Animation in the Dopesheet on page 241.

You can choose to either scale and offset using explicit values, or else you can retime an animation by fitting it into a specified frame range. For example, you can easily retime an animation that used to go from frame 1 to 100 so that it now goes from frame 30 to 50. You can also reverse an animation easily: for example, instead of going from frame 1 to 100, it goes from frame 100 to 1.

To offset, scale, or retime animation

1. Do one of the following, choosing the corresponding Animation > Sequence Animation command from the Animation panel:

   - To offset, scale, or retime animation for only marked parameters, select the objects and mark the parameters whose animation you want to change, then choose Sequence Animation > Marked Parameters.

     For information on marking parameters, see Marking Parameters for Animation on page 35.

     or

   - To offset, scale, or retime all animation for selected objects, select the objects whose animation you want to change and choose Sequence Animation > All Parameters on Selection.

     or

   - To offset, scale, or retime all animation in the scene, make sure that nothing is selected and choose Sequence Animation > All Scene.
2. In the Sequence dialog box that appears, select the appropriate **Input Method**:
   - Select **Offset and Scale** to either offset or scale the animation.
   
   Then follow the steps in *Offsetting or Scaling Animation* on page 56.
   
   or
   
   - Select **Retime (Before/After Range)** to retime the animation.
   
   Then follow the steps in *Retiming Animation* on page 57.

### Offsetting or Scaling Animation

**To offset or scale the animation**

1. Select **Offset and Scale** as the Input Method in the Sequence dialog box.

2. With the **Offset**, specify the number of frames by which to offset the animation.

3. Enter a **Scale** value by which to stretch or shrink the animation.

   This value is the number by which the current length of the scene (or frame range) is multiplied. For example, using a value of 3 here makes a 100-frame scene 300 frames long.

   When you scale you can also choose to **Pivot** the animation at a certain frame, which lets you reverse an animation at that point.

The main timeline’s start and end frames are not updated automatically if the resulting scaled/offset/retimed animation goes outside the current frame range.

To view your entire animation, you must either reset the start and end frames on the timeline or open the animation editor to view the scaled or offset function curves.
4. Specify the frame range you want to affect by setting the Start and End Frame values for the Affected Scale & Offset Range.

5. Select the Affected Items you want to scale or offset from the Sequence Mode list. You can choose from function curves, clips in the animation mixer, or all animation sources, as well as a combination of these types.

Retiming Animation

You can retime an animation by fitting it into a specified frame range. For example, you can easily retime an animation that used to go from frame 1 to 100 so that it now goes from frame 30 to 50. You can also reverse an animation easily: for example, instead of going from frame 1 to 100, it goes from frame 100 to 1.

To retime an animation

1. Select Retime (Before/After Range) as the Input Method in the Sequence dialog box.

2. Specify the Retime Range Before values, which are the frames of animation you want to use for retiming.

3. Specify the new frame range in which you want your animation to fit by setting the Retime Range After frame values.
4. Select the **Affected Items** you want to retime from the Sequence Mode list. You can choose from function curves, clips in the animation mixer, or all animation sources, as well as a combination of these types.

Selected fcurve has been **retimed** so that a range of 125 frames in the middle of the sequence has been compressed to into a range of 80 frames. The ghosted fcurve shows the original fcurve’s size and shape.
Plotting Animation

When you plot the animation on an object, the animation is evaluated frame by frame and function curves are created. Plotting is useful for generating function curves from any type of animation, such as from constraints. You can, for example, plot the animation of a constrained object and then remove the constraints so that only the plotted animation remains on the object.

Animation of an object constrained between two points is plotted.

You can choose the type of function curve you want to create when you plot the animation: standard (with different types of interpolation), raw, Boolean, or integer.

In addition to plotting general animation, you can also plot shape animation (see Plotting Shape Animation on page 89 in the Shape Animation guide) or plot a curve from an object’s animated transformation values (see Creating a Curve by Plotting an Object’s Transformation on page 274).

You can use ghosting to view an object’s plotted animation as a motion trail—see Ghosting Animated Objects on page 42.

Keeping or Deleting the Resulting Action

Plotting is done by first creating an action source (see Action Sources and Clips on page 90 in the Nonlinear Animation guide for information). Depending on the settings you select in the Plot dialog box, you can choose to keep or delete this action source after the animation is plotted:

- You can apply the plot function curves immediately to the object and delete the action source. This gives the same results as plotting does in SOFTIMAGE|3D.

or

- You can apply the plot function curves to the object and also keep them stored in an action source. This may be useful if you’re using the animation mixer.

or
You can keep the action source of the plotted function curves but not have it applied to the object immediately. You must then select the source and apply it to the object using the Actions > Apply > Action command (see Restoring and Applying Animation from Action Sources on page 119 in the Nonlinear Animation guide).

You can then load the action source into the animation mixer to make a clip of it, as you would for any other action source.

If you’re using action sources in the animation mixer, you can use plotting to “bake” animation from any combination of sources (including other action clips) into a single action source. For example, you can combine the animation of a walk cycle and an arm waving into a single action clip instead of creating a compound clip.

Tips for Plotting

While the process of plotting is fairly straightforward, here are a few tips to make sure that you’re plotting what you want.

Plotting Animation for Children Objects

If you branch-select an object, animation will not be plotted for its children. This is unlike storing actions, which automatically expands the branch selection; rather, it is similar to saving keys that work only on the root of the selection.

To plot the animation of an object’s children as well, branch-select the object and then choose Select > Select Child Nodes.

Marking Parameters

If you are not plotting local transformations of any type or constrained parameters, make sure to mark the parameters whose animation or values are to be plotted and then choose the Plot > Marked Parameters command.

For example, if you want to plot an object’s global transformation parameters, you must mark them first and then choose this command. You can also create a marking set for the object with the appropriate parameters in them and then plot with this command.

For more information, see Marking Parameters for Animation on page 35.

Plotting Blended FK/IK Animation on a Chain

If you want to plot the blended FK/IK animation on a chain, you need to mark the correct parameters to get the correct results.

• To plot FK, mark the bones’ local rotation parameters.
• To plot IK, mark the effector’s local position parameters.

Then choose the Plot > Marked Parameters command.
Plotting Animation

**Plotting the Global Position of Effectors for SOFTIMAGE|3D-type Skeletons**

When you are working with a SOFTIMAGE|3D-type skeleton, you usually key the effector's global position (translation). However, the Plot > Translation command plots an object's local position, not global.

To plot the effector's global position, mark its global translation parameters and choose the Plot > Marked Parameters command.

For more information, see *Marking Parameters for Animation* on page 35.

**Plotting the Animation**

You can plot any type of animation on an object or model.

If you have multiple objects selected in different models (including submodels), only the objects in the same model as the first object you selected will be plotted. This is because only one model can be evaluated at a time.

**To plot animation**

1. Select one or more objects or a model whose animation you want to plot.
2. Mark the selected object’s parameters, if necessary.
3. From the Tools > Plot menu of the Animate toolbar, choose the appropriate command for the type of animation you want to plot:
   - **All Transformations** plots only the **local** transformation (scaling, rotation, and translation) parameters. All local transformation parameters are plotted, whether they are animated or not.
   - **Positions** plots only the **local** position (translation) parameters.
   - **Rotations** plots only the **local** rotation (orientation) parameters.
   - **Marked Parameters** plots only the parameters that you have marked.
   - **Constrained Transforms** plots only transformation parameters that are affected by constraints (for example, if a direction constraint is applied to an object, only its rotation parameters would be plotted). Only active constraints are plotted.

When the plotting is finished, a message box opens displaying the list of constraints that were plotted. You can then choose to remove these constraints from the object so that you are left with only the plotted animation.
4. In the Plot dialog box, specify the following information:

Set the frame range and step of animation to be plotted.

Select the type of fcurve to be plotted.

Choose what to do with the resulting plotted animation and the action source.

The settings you make in this dialog box are kept so that the next time you plot (within the same session of XSI), the same options are selected. This makes it easy to do several similar plots in a row.

These settings are kept in a custom parameter set named PlotToAction that is found under the scene root.

To give a name to the action source

- Specify the Action Name for the new action source that will be created. This only applies if you are going to keep the resulting action source (deselect the Delete plotted action option).

To set the time span and frame step

- Set the Start Frame and End Frame for the range of frames to be plotted.
- Set the Step Value to specify the frame step increment. If this value is 1, every frame is evaluated and a key is set at every frame; if this value is 2, every other frame is evaluated and a key is set at every other frame.

To keep and apply or delete the plotted action

- Select Apply plotted animation to object to apply the plotted action on the selected objects right away.

If you deselect this option, the action source is created (with the name you specified) in the model’s Mixer > Sources > Animation folder and the Sources > Model folder at the scene level. You must then select the source and apply it to the object by choosing the Actions > Apply > Action command on the Animate toolbar.
• Select Delete plotted action to do just that: the plotted action source is not kept.

If you deselect this option, an action source is created with the Action Name that you specified and you must apply it to the object, as previously described.

If you select both Apply plotted animation to object and Delete plotted action, the plotted animation is applied to the object but no action source is kept. This gives the same results as plotting does in SOFTIMAGE|3D.

To paste keys on the original fcurve rather than replace it

• Select Apply using paste keys rather than replace curve to paste the keys from the plotted animation on the object’s original fcurves rather than replacing the entire fcurve with the plotted one. This behavior is more like SOFTIMAGE|3D.

The pasted keys don’t need to be of the same type as the fcurve to which they are added. For example, you can paste Raw Data keys to a Standard fcurve.

If you deselect this option, the object’s original fcurve is replaced by the plotted fcurve.

5. Click OK to create the plotted function curves for the animation.

6. If you chose Plot > Constrained Transforms, a message box opens displaying the list of constraints that were plotted. You can then choose to remove these constraints from the object so that you are left with only the plotted animation.

Options for the Resulting Plotted Function Curves

You can choose the type of function curve that is created when you plot the animation: standard (with different types of interpolation), raw, Boolean, or integer. There are also continuous rotation and fitting options.
To select the type of resulting function curve

- Select one of these options for Kind of Fcurve:

  - **Standard** maps time and values—the same type of fcurve as is used in the animation editor.  
    With the Standard fcurve, you can select one of these Interpolation types: Spline, Linear, or Constant. These are the same as fcurve interpolations in the animation editor (see Choosing a Function Curve Interpolation Type on page 210). If you plan to do curve fitting, it's best to create standard spline curves.

  - **Boolean** fcurves map time to Boolean values (true or false). These create staircase-like function curves, useful for plotting animated toggles (on/off values).

  - **Integer** fcurves map time to integer values. These also create staircase-like fcurves, useful for plotting the values of integer parameters, such as RGB values.

  - **Raw Data** maps time and values, like standard fcurves but using only linear interpolation. They require less memory, which means a smaller file size; useful when long animation sequences are plotted, such as from motion capture sessions. Raw data is useful when plotting every frame without fitting, as with motion capture data.

To process continuous rotations

- Select **Process rotations to ensure continuity** to treat orientation as a single entity instead of individual X, Y, and Z rotation function curves. This allows the plot processing to avoid "spikes" in the continuity of the rotation curves, when they may jump between equivalent (but discontinuous) representations.

  You may want to select this option when plotting rotations if you plan to stretch, mix, or manipulate them as actions in the mixer.

To fit the keys on the fcurve

Curve fitting is processed after the continuous rotation is processed.

1. Select **Fit Plotted Values with an Fcurve** to reduce the number of keys on the fcurves resulting from the plot.

2. Set the **Fit Tolerance** value to adjust the closeness of the fit of the curve. Smaller values respect the original shape of the curve more by using more keys, resulting in a fit that is closer to the original curve.
Muting (Deactivating) Animation

When you mute animation, you temporarily disable the effect that certain animated parameters have on your objects. This helps you test out different factors of an animation as you’re working. You can mute any type of animation in XSI.

There are many ways in which you can mute animation:

- You can mute function curves in the fcurve editor—see Deactivating (Muting) Function Curves on page 236.
- You can mute animation (function curves, constraints, expressions, and deform operators) using the dopesheet—see Deactivating (Muting) Animation in the Dopesheet on page 261.
- You can mute animation in the keying panel—see Keying Parameters in the Keying Panel on page 109. The animation icon turns white when the animation is muted.
- You can deactivate individual constraints or expressions: see Activating and Deactivating Constraints on page 324 and Activating and Deactivating Expressions on page 364.
- You can mute all local transformations and constraints by right-clicking on the Local Transform and Constraint icons in the explorer and deselecting Active. A little orange “M” appears on the icons to remind you that they are muted.
- You can mute the effect of scaling, rotation, and translation individually from the Options page in the Local Transform property editor (select an object and press Ctrl+k). If you deselect the Active option at the top of the page, all transformations are muted. For more information on these options, see Animating Transformations in Hierarchies on page 148.
- If you have made action clips for the animation, use the mute buttons (the m button) for any mixer track to mute all action clips on that track (see Working with Tracks on page 28 in the Nonlinear Animation guide). You can also mute individual clips (see Working with Clips on page 32 in the same guide).
Checking for Animation Dependency Cycles

Building animation relationships involving constraints, expressions, and scripted operators can often lead to evaluation cycles. An evaluation cycle is a looped graph dependency where an operation becomes dependent upon itself.

As a simple example, constraining object A to object B (such as with an orientation constraint) then constraining object B to object A with the same type of constraint will create a cycle of dependencies. Another example is putting an expression on the Length parameter of a bone in a chain, where the expression is the distance from the chain’s root to its effector. This is a cycle because you’re driving the Length parameter using the translation of the effector, but the effector’s position is bound to the bone’s length (and rotation).

XSI implements internal cycle checking to prevent cycles from occurring in a scene. Using the CycleChecking command, XSI checks for cycles as soon as you create a constraint or expression, and warns you immediately if a cycle in dependencies is created.

The CycleChecking command simply reports cycle details in the command history so that you can check that there’s not a problem; for example, sometimes you want to intentionally create a cycle, such as direction-constraining two objects to create a piston. This command does not prevent a scene from loading even if there is a cycle problem.

If there is a problem cycle, the warning messages and information in the script editor can help you break it.

Avoiding Cycles

To help you prevent creating cycles, you can do any of the following:

- Check where constraints already exist in a 3D view or in the schematic view—see Viewing Constraints and Their Information on page 288.
- Select an object and choose Constrain > Select Constraining Objects to see the objects to which the selected object is constrained.
- Type the command CycleChecking in the script editor and click Run to check an entire scene. Select the command and press F1 to get more information on it.

For information on constraints, see Chapter 10: Animating with Constraints on page 283.

For information on expressions, see Chapter 12: Animating with Expressions on page 345.

For information on scripted operators, see Scripted Operators on page 151 in the Customization guide.
Disabling Cycle Checking

The Disable cycle checking when applying animation operators option on the Scene Debugging tab of the Data Management preferences suppresses checking for evaluation cycles when creating constraints and expressions. This may be useful for avoiding unwanted warning messages in some types of rigs where cycles are desired. Note that cycle checking is always performed when applying topology operators, connecting weight maps, and moving operators.

Breaking Cycles

To break an evaluation cycle, you should look for and remove the main contributors within the cycle.

1. Open the script editor and check for cycle warning messages and information to help you break a cycle.

   In this example, these warning messages appeared because there was a cycle created by two expressions driving the same parameter.

2. To look for key contributors in the cycle, start with the Cycle Breaking Point statement, which is the location where the evaluation check has terminated. This is the critical location for scene evaluation differences or any odd results and behaviors you may notice with problematic cycles.

3. Trace down the output with the Cycle through operators until the same Cycle Breaking Point operator is found and labeled as a Cycle through operator.

   These are the main contributors to the cycle: breaking one of these key dependencies will break the dependency cycle.

4. Look through the contributors for familiar cycle creators such as Expression, ScriptedOp, or kine.Constraints.

5. Check your animation that uses these operators and resolve the problem by reworking the animation or deleting the operator.

Tips

- If a cycle exists with constrained objects, a message appears telling you to remove the last constraint you applied to resolve the conflict (see Removing (Relaxing) Constraints on page 325).
• You can also deactivate a constraint (see *Activating and Deactivating Constraints* on page 324) to break a cycle because inactive constraints are not considered when XSI checks for cycles.

• Muting or having 0 as the Blend Weight value for a constraint doesn’t stop it from being evaluated so it could contribute to a cycle.
Removing Animation

You can remove animation from selected objects or marked parameters using any of the numerous commands from the Animation menu in the Animation panel. As well, using the explorer or a property editor, you can easily remove animation from individual parameters.

To remove keyframed animation specifically, see Chapter 4: Animating with Keys on page 101.

You can also use the animation editor to delete keyframes (key points) on the function curve (see Deleting Keys on page 197), delete keys in the timeline (see Deleting Keys in the Timeline on page 137), or use the dopeshot to remove keys (see Cutting, Copying, and Pasting Regions on page 256).

Using the Animation Menu

To remove animation from an object

1. Select the object or mark the parameters from which you want to remove animation.

2. From the Animation panel, choose Animation > Remove Animation and one of these commands:

   - Choose from Marked Parameters to remove all keys (fcurves) from the selected objects’ marked parameters.

   - Choose from Transforms to remove animation from all local scaling, rotation, and translation parameters of the selected objects. You can also choose to remove each of these types of local transformations individually (X, Y, and Z parameters for each type) with their appropriate commands.

   - Choose from All Parameters, Any Animation Type to remove all animation from all parameters on selected objects. This includes fcurves, scripted operators, expressions, linked parameters, etc.

   - Choose from Objects, Constraints to remove all constraints from the selected objects. You can also use the Constrain > Remove All Constraints command on the Constrain panel to do this.

   - Choose from All Parameters, Expressions to remove only expressions from all parameters on selected objects.
Chapter 2 • Tasks for Any Type of Animation

- Choose from All Parameters, Fcurves to remove only fcurves from all parameters on selected objects.

- Choose from All Parameters, Static Fcurves to remove fcurves that have no animation on them anymore (the fcurves are flat or static) from all parameters on selected objects.

- Choose from All Parameters, Scripted Operators to remove only scripted operators from all parameters on selected objects.

Using the Explorer or a Property Editor

Another way in which you can remove animation is at the parameter level using the explorer or property editors. This can be animation from keys, linked parameters, expressions, or scripted operators.

To remove animation from parameters

1. Display the property editor from which you want to remove animation.

2. Mark the parameters whose keys you want to delete.

3. Do one of the following:
   
   - Right-click the keyframe icon in the property editor and choose Remove Animation.
   
   or
   
   - In the explorer or a property editor, right-click the animation icon of the parameter whose animation you want to remove and choose Remove Animation.

   ![In a property editor, right-click the keyframe icon and choose Remove Animation to remove animation from all or marked animated parameters on the property page.](image1)

   ![In either the explorer or a property editor, right-click the animation icon of an animated parameter and choose Remove Animation.](image2)
Chapter 3  Playing Animation

Although playing an animation is as simple as clicking the Play icon, there are a number of options that first need to be set up for a scene.

The first thing you need to do before even starting an animation is to set up your frame rate and format to match the medium in which you will be saving the final animation. See Setting the Scene’s Frame Format and Rate on page 73 for more information on this.
A big part of the animation process is the constant tweaking and replaying of the animation to see that you get things right. There are a number of tools in XSI that help you do this, as described in Playing the Animation on page 77. As well, Optimizing Playback on page 89 gives a number of tips on how you can make the playback go as quickly as possible.

In addition to playing the animation in a viewport, you can also capture images from an animation sequence and play them back in a flipbook. See Capturing Animation in a Viewport on page 91 and Previewing Animation in a Flipbook on page 94 for information on these.
Setting the Scene’s Frame Format and Rate

In animation, the smallest unit of time is the amount required to display a single frame. The speed at which frames are displayed, or the frame rate, is always determined by how the final animation will be viewed.

For example, in North America, television plays at 30 fps (frames per second) with the NTSC format, while in most of Europe it plays at 25 fps with the PAL format. Film, on the other hand, plays at 24 fps. Most traditional animation is shot “on twos,” which means each individual image is held for two frames. So, practically speaking, film animation shot on twos runs at 12 fps, though the film itself plays at 24 fps.

If you are compositing your animation with other film or video footage, it’s usually best for the animation to be at the same frame rate as the footage.

When you change the timing of the animation, you change the way that the actions look. This means that the timing that looked correct while you were previewing it in XSI may not look as good on video or film. For example, an action that spans 24 frames would take one second on film; changing the frame rate to suit North American video at 30 fps would cause the same 24 frames to span 0.8 seconds. As a result, you should make sure to preview your animation at the same frame rate that will be used in the final format.

Setting the Default Frame Rate and Format

You can set up the default frame rate and format for your scene using the options in the Output Format property editor.

These settings are propagated to the playback (interaction) frame rate and format options in the Play Control property editor (see Setting the Playback Frame Rate and Format on page 75), as well as to the options in the Capture Viewport dialog box (see Capturing Animation in a Viewport on page 91). Of course, you can override the settings in those property editor with different settings, if you like.

To set the scene’s default format and frame rate

1. Choose File > Preferences from the main menu.
2. In the explorer that appears, select Output Format or click its icon to open the Output Format property editor.
3. Select a Frame Format and Frame Rate in the Default Frame Rate area.
The frame format you select should match the intended final destination of your animation. For example, you would use 24 fps for film, 29.97 fps for NTSC video, and so on.

The default Frame Format is NTSC at 29.97 frames per second. You can also choose PAL, Film, 30fps, or specify a Custom frame rate.

4. If you select Custom as the Frame Format, you must specify its Frame Rate. For example, if you want a frame rate of 12.5, enter 12.5 in the Frame Rate text box.

5. Set the Frame Step to determine how many frames you want to skip. By default, the Frame Step is set to 1, meaning that each frame in the scene is played back. If you wish to view every other frame during playback, set this value to 2; if you wish to view every eighth frame, set this value to 8, and so on.

6. Select the Update Play Control option if you want the playback settings in the Play Control property editor to be automatically updated.
Setting the Playback Frame Rate and Format

The playback frame rate and format are used only for the playback interaction in XSI. These playback options take the default settings that you have made in the Output Format property editor (see previous section), but you can set them as you like for playback without affecting the default frame format and rate set for the scene.

To set the playback frame rate and format

1. Choose Playback > Playback Options in the Playback panel to open the Play Control property editor.

2. On the Options page in the Play Control property editor, select the Frame Format you want.

   If you create image sources for texture sequences, make sure that this frame rate and the scene's default frame rate match.

   You can choose from NTSC at 29.97 frames per second, PAL, Film, 30fps or specify a Custom frame rate.

   **Frame Step** determines how many frames should be skipped when playing back your animation.

   The **Frame Format** should match the intended final destination of your animation. If the frame rate you want isn’t listed, select Custom frame rate as the format and enter the number in the Frame Rate text box.

   These options let function curves maintain their keyframes at their original frame values when you change the frame rate.
3. If you select Custom as the Frame Format, you must specify its Frame Rate. For example, if you want a frame rate of 12.5, enter 12.5 in the Frame Rate text box.

4. Set the Frame Step to determine how many frames you want to skip.

   By default, the Frame Step controls in the Play Control property page are set to 1, meaning that each frame in the scene is played back. If you wish to view every other frame during playback, set this value to 2; if you wish to view every eighth frame, set this value to 8, and so on.

   The increment icons (next and previous frames) on the Playback panel are set by the value for Frame Step. Frame Step does not work when using real-time playback.

Displaying the Current Frame Rate

   To see the rate at which your animation is being played back

   • Click the eye icon in a viewport and choose Frame Rate
   
   or
   
   • Press Ctrl+r and click in a viewport.
   
   or
   
   • Click the eye icon in a viewport and choose Visibility Options (or press Shift+s). On the Stats page in the Camera Visibility property editor, select Show Frame Rate.

   The displayed rate is updated during playback.
Playing the Animation

The main way of playing back animation is by dragging the playback cursor on the timeline or by using the playback controls below the timeline. You can also set the time range for playing back, play back in real time, and optimize the playback in different ways.

Before you start playing back the animation, you should set up the time range, the time display format, and the timeline’s start and end frames.

Setting the Time Range for Playback

The time range lets you do two things:

- Set the global start and end frames of the scene.
- View and play back a smaller range of these frames in the timeline. For example, if you are working with an animation sequence that is very long, it’s useful to be able to focus on a smaller range of frames for specific tasks.

To display or hide the time range

- Choose View > Optional Panels > Time Range Slider from the main menu.

The time range appears below the timeline, with the global start and end frames displayed at either of its ends and a time range slider in the middle.

To set the overall time range displayed

Do either of the following:

- Enter values in the Start and End frame boxes at either end of the time range.

  or

- Set frames for the Global In/Out parameters in the Play Control property editor (choose Playback > Playback Options in the playback panel).

  or
Chapter 3 • Playing Animation

- Set the default preference for the Global In/Out parameters in the Animation Preferences property editor (choose File > Preferences from the main menu).

To change the range of frames displayed in the timeline
Changing the values of the time range slider determines which frames are displayed in the timeline.

- Drag the range slider’s left or right handles to adjust the length of the frame range (compress or extend the range). Doing this updates the timeline’s start and end frames.

- Drag in the middle of the range slider to shift the time range without changing the range’s length. Doing this updates the timeline’s start and end frames.

Setting the Scene’s Start and End Frames
Before you play back an animation, you should set the appropriate start and end frames for the scene. These define the range of frames that you can play in the scene.

To set the scene’s playback frames
Do either of the following:

- Set the frame numbers in the Start and End Frame boxes at either end of the timeline.

  or

- Set the Frame In and Frame Out values in the Play Control property editor (choose Playback > Playback Options in the playback panel).

  These values must be within the range of frames specified for the Global In/Out values (time range) in the same property editor.

Setting the Timeline Display Format
By default, the timeline is set to display frames but you can change this to display other options. When you change these settings, they also affect the timelines found in the animation editor (fcurve editor and the dopesheet) and the animation mixer.
Playing the Animation

To set the timeline display

1. Choose File > Preferences from the main menu.
2. In the explorer that appears, select Display or click its icon to open the Display property editor.
3. Click the Time Display Format tab and select either of these options:
   - Select Display As Frames to do just that.
   - or
   - Select Use Custom Display Format, then select a format from the Display Format list.

Playing Back in the Timeline

The timeline is used to display the current playback position of the animation and to manually move between different frames.

The current frame (or time position depending on the format) is indicated by the position of the playback cursor in the timeline, the frame number beside the playback cursor, and in the current frame box in the Playback panel.

The red playback cursor indicates which frame is currently displayed in the viewports. The frame number is displayed beside the cursor as well as in the current frame box beside the playback controls.

- Left-clicking anywhere in the timeline changes to the frame. You can also scrub (drag) the playback cursor back and forth (see Scrubbing the Animation on page 83 for different scrubbing options) in the timeline.
- Middle-clicking+dragging the playback cursor moves to the selected frames but does not update the scene. The cursor and frame number beside it (and in the current frame box) turn green. To refresh the scene, click the cursor.

By middle-clicking+dragging a frame to another frame, you can copy the current values at this frame to any number of other frames and then set keys. For more information on this, see Keying the Current Values at a Different Frame on page 135.
Right-clicking in the timeline opens the context menu used for editing keys in the timeline. For more information, see Editing Keys in the Timeline on page 125.

To hide or display the timeline

- Choose View > Optional Panels > Timeline from the main menu to toggle the display of the timeline.

Displaying the Current Frame Number

To display the current frame number in a viewport

1. Click the eye icon in a viewport and choose Visibility Options.
2. On the Stats page, select Show Current Time.

This uses the settings defined in the Output Format Preferences property editor.

If you want to capture the animation in your viewport (see Capturing Animation in a Viewport on page 91), this option is useful for having the frame number appear on each captured frame.

Using the Playback Controls

The controls in the Playback panel below the timeline allow you to view and play animation and audio in a variety of ways:

To play forward

- Click the Play icon to play from the first frame on the timeline.
  
or
- Middle-click the Play icon to play forward from the current frame.
  
or
- Press the up-arrow key to play forward from the current frame.
To play backward

- Click the Play Backward icon to play from the last frame on the timeline.
  
  or

- Middle-click the Play Backward icon to play backward from the current frame.

To stop the playback

- Click or middle-click the Play or Play Backward icon again.
  
  or

- Click any mouse button anywhere in the viewports or timeline
  
  or

- Press the down-arrow key.

To go to the first/last frame of the timeline

- Click the First/Last Frame icon.
  
  or

- Press the Home/End key.
  
  or

- Choose Playback > First Frame/Last Frame.

To move forward/backward one frame

- Click the Previous/Next Frame icon.
  
  or

- Press the left/right-arrow key.
  
  or

- Choose Playback > Next Frame/Previous Frame.

To repeat the animation in a continuous loop

- Click the Loop icon. When you do this, two yellow bars appear in the timeline at the first and last frames to indicate the start and end frames of the loop (see Looping the Playback on page 82).

To execute a script whenever a frame is displayed

- Choose Playback > Playback Options and click the Update tab. Then enter the script name in the Command box.
Chapter 3 • Playing Animation

Looping the Playback

With looping, you can play back the animation in a continuous loop. This can be handy when you want to test out a small area of the animation but don’t want to play back all of it.

There are different ways in which you can loop the animation, depending on how you’re working: you can set up loop markers in the timeline to delimit a range of frames or you can use the Quick Loop mode.

To activate looping

- Click the Loop icon in the Playback panel.

or

- Select the Loop option in the Play Control property editor (choose Playback > Playback Options in the playback panel).

Two yellow bars appear at either end of the timeline to indicate the start and end frames of the loop.

Yellow loop markers define the range of frames to be repeated in a continuous loop when you play the animation.

Setting the Looping Range

To set the loop playback range

- Drag the yellow loop markers at either end of the timeline to set the loop range, or Ctrl+click in the timeline at the frames where you want them.

or

- Set the Loop In and Loop Out values in the Play Control property editor.

By default, the loop in and out frames match the scene’s start and end frames. If the loop range matches the scene’s range and you change the scene’s start or end frame, the loop range automatically updates to match the scene’s new frame range.

Using Quick Loop

To quickly loop a range of frames around the playback cursor

1. Do either of the following:
   - Choose Playback > Playback/Audio Preferences from the playback panel at the bottom of the XSI window.

   or

   - Choose File > Preferences from the main menu. In the Preferences explorer that opens, select Interaction or click its icon.
2. In the Interaction Preferences property editor, click the Playback and Audio tab and select **Enable Quick Loop Mode**.

3. Set the number of **Frames Before/After Cursor**. This determines where and for how long you want the loop.

   For example, you could set 1 for the Frames Before Cursor and 20 for the Frames After Cursor to loop the 20 frames of audio that occur after the position of the playback cursor in the timeline.

4. Move the playback cursor in the timeline to where you want to loop the animation and hold down the mouse button.

   As long as you hold down the mouse button, the animation is looped back around the playback cursor according to the frames you have set for the Quick Loop mode.

5. When you release the mouse button, the looping stops. You can then click anywhere else in the timeline to continue to quickly loop other parts of the animation.

**Scrubbing the Animation**

Scrubbing means to drag the playback cursor back and forth in timeline. This is something that you need to do frequently as you check the animation in a segment. There are different modes that you can set for scrubbing, as well as the scrubbing playback rate.

*To set options for scrubbing*

1. Open the Interaction preferences property editor (see above) and click the Playback and Audio tab.

2. Set the options in the **Scrubbing** area:

   - **Default Scrub Mode** is the main method to use when scrubbing:
     - **Direct Seek**: Plays exactly where you scrub the playback cursor.
Chapter 3 • Playing Animation

- **Play User Rate**: Plays at the speed you specify for the Scrubbing > User Rate percentage value (below).
- **Play Realtime**: Plays at realtime speed.
- **Play Vari Speed**: Plays according to the speed at which you scrub the playback cursor.

- **Alternate Scrub Mode** is a different scrubbing method you can use when you press the Alt key. For example, you may want to scrub over only one area with a specific rate of playback (Play User Rate), but you want to do the main scrubbing with a different setting (such as Play Realtime).

The options for this parameter are the same as for Default Scrub Mode.

- **User Rate %** is the playback speed when you select Play User Rate for the Default and/or Alternate Scrub Mode options. This works in the same way as the Realtime Playback > User Rate % option.

See Audio and Animation on page 199 in the Nonlinear Animation guide for more information on the audio options.

3. Drag the playback cursor in the timeline to scrub the animation using these settings.
Playing Back All Frames or Playing in Real Time

XSI can play all frames of an animation or it can play the animation in real time. It accomplishes real-time playback by choosing the frames to display based on the elapsed time.

For instance, if it takes half a second to display the first frame in a 24 fps animation, frames 2 through 12 will be skipped and frame 13 is displayed next. In other words, when it comes time to display a frame, the frame that should be shown at that moment is displayed.

To play back all frames
To see all the frames in your animation (though not necessarily in real-time), do one of the following:

- Choose Playback > Play All Frames.
  or
- Toggle the All/RT button to All.

To play back frames in real time
Do one of the following:

- Choose Playback > Real-Time Playback.
  or
- Toggle the All/RT button to RT.
  or
- Press Shift+up-arrow key.

Setting the Realtime Playback Rate

The frame rate determines how smoothly the animation plays back. You can set the realtime playback rate to whatever percentage you like.

To set a rate preference for realtime playback

1. Do either of the following:
   - Choose Playback > Playback/Audio Preferences from the playback panel at the bottom of the XSI window.
     or
   - Choose File > Preferences from the main menu. In the Preferences explorer that opens, select Interaction or click its icon.
2. On the Playback and Audio tab, set the Realtime Playback > User Rate % to any positive value. 100% is full speed, while 50% is half speed, 200% is double speed, and so on.

The User Rate % slider goes only to 200 but you can enter values higher than that.

**To select a predefined realtime playback rate**

- In the Playback menu, you can quickly select predefined realtime playback speed: 75%, 50%, or 25%. These options are available only when realtime playback is active.

  When you do this, the User Rate % value in the Interaction preferences property editor is updated to match what you have selected.

You may want to use the Fast Playback option in conjunction with real-time playback to speed up the frame rate. The Fast Playback option caches the raw geometry of objects at each frame the first time you play an animation, then plays it back as quickly as possible. This allows XSI to achieve very fast playback rates, which can help you evaluate an animation’s motion and timing more accurately.

Fast playback is different from capturing images in the viewport (see *Capturing Animation in a Viewport* on page 91), which stores an image at each frame.

**To activate and set up fast playback**

1. Do either of the following to activate fast playback:

   - Select Fast Playback from the Display Type menu in a viewport.

   or

   - Choose Display Options from the Display Type menu to open the Display Options property editor. Click the Performance tab and select the Fast Playback option.
2. Set the Fast Playback options on the Performance page.

- If you select Wireframe Capture Only, you can select Use 3D Cache. 3D caching stores extra depth information for the captured geometry so that you can orbit, pan, and zoom the camera during playback. 2D caching (when Use 3D Cache is not selected) stores the data in a more compact form to save memory.

Fast playback captures only the visible geometry, so if the camera is zoomed out when you play back a 3D cache, the geometry that was outside the view at the time of capture is not visible.

- Set the Cache Size value, which is the amount of RAM allocated per frame when caching images.

When you have Fast Playback on, you may get a “cache overflow” message. Make sure to either deselect Fast Playback or increase the Cache Size.

3. Play the animation.

The first time you play the animation, each frame is cached. When you play back the animation again, it is played as quickly as possible.

If you change the animation after you cache it, the cache will not be updated automatically. You must deactivate and then reactivate the Fast Playback option to accommodate the changes, or click the Clear button on the Performance property page to clear the cache.
Setting the Playback Speed Limit

Sometimes things can go just a little too fast. For example, when playing back all frames, it is not really useful to have the animation play back faster than the set frame rate, or you want to play all frames to cache them, but the playback is too fast. Luckily, there is a simple remedy for limiting the playback speed to the scene’s current frame rate.

To prevent all frames from playing back too quickly

1. Do either of the following:
   - Choose Playback > Playback/Audio Preferences from the Playback panel.
   - Choose File > Preferences from the main menu. In the Preferences explorer that opens, select Interaction or click its icon.

2. Select the Limit Max Playback Rate to Scene Frame Rate option on the Playback and Audio page.

This prevents the All Frames playback option (see previous page) from playing back faster than the scene’s playback frame rate.
Optimizing Playback

The more you can see of the animation's essentials, the better the visual feedback you have to work with. Ultimately, this helps you produce better animation.

You may not always be able to play back each frame at the correct rate, however. At every frame, XSI must check if positions, orientation, scaling, geometry, visibility, and many other animatable parameters have changed. If they have changed, XSI must then calculate the extent of the change, calculate the effects of the change, and then redraw the new image. The more calculations to be done, the longer it will take to display each frame.

If playback is too slow, you must decide what aspect of the animation is critical for display and eliminate the extraneous elements that may be limiting the playback speed. For example, if you’re just verifying that your phantom’s coloring is shimmering properly, you can display the shifting color but not the phantom’s translation. Since there is less to calculate, the scene will play back faster—especially if the phantom’s animation was very complex.

Some factors that slow down playback are inevitable. For instance, the more objects in a scene, the slower the playback, but even that can be minimized. However, most causes can be easily remedied, such as:

- Other tasks running on your computer.
- The number of active viewports. Muting non-essential viewports speeds up playback in the remaining viewports. To do this, middle-click the viewport’s letter (A, B, C, or D) so that it turns orange or right-click it and choose Mute. You can also hide the viewport’s grid (press g).
- The method of displaying models in the viewports. Using options such as textures and shaded models significantly slows playback. For more information, see Display Types on page 156 of the Interface and Tools guide.
- Setting a number of display options as discussed in Setting Display Options on page 156 of the Interface and Tools guide.
- Put objects you don’t currently need in layers and set them to be not visible (or not renderable, if you are previewing in the render region). For more information, see Layers on page 155 of the Scene Elements guide.
Choosing What’s Updated During Playback

You can also control which animatable parameters will be updated during playback.

To choose what’s updated during playback

- Choose Playback > Playback Update Mode from the Playback panel, then select an option from the menu:
  - Update All updates all scene elements during playback.
  - Selected updates only selected scene elements during playback.
  - Selected + Unselected SRT updates only the selected element, as well as elements with animated scaling, rotation, or translation parameters.
  - SRT/Selected SRT updates only elements/selected elements with animated scaling, rotation, or translation parameters during playback.
  - Geometry/Selected Geometry updates only geometric objects/selected geometric objects during playback.
  - Custom opens the Play Control property editor in which you can click the Update tab and specify more detailed options.

By specifying update parameters for selected objects and near or far unselected objects, you can visualize exactly what you need while eliminating unnecessary animation.

You can decide what kind of animation should be displayed for these objects: all, none, SRT (transformations), geometry, other, or any combination.

Any objects that fall within this distance from the camera (in Softimage units) are considered near. All other objects are considered far.
Capturing Animation in a Viewport

You can cache each frame of an animation in an image file and create a flipbook. A flipbook is any sequence of cached images saved to disk.

Anything that is shown in the viewport you choose is captured—render region, rotoscoped scene with background, or any display type (wireframe, textured, shaded, etc.). For example, you may want to set the display type to Hidden Line Removal for a “pencil test” effect.

If there is a render region in the viewport, whatever is shown in that area is what is cached, not the whole viewport.

If you have any audio files in your scene, you can also capture sound along with the images (see Capturing Animation with Sound on page 93).

- To display the frame number in the viewport, press Shift+s. On the Stats page in the Camera Visibility property editor, select Show Current Time. This uses the settings defined in the Output Format Preferences property editor.
- In the Render Options property editor, use OpenGL render engine for higher quality captured images—see Rendering Options on page 55 in the Rendering guide for more information.

These cached images can be loaded and quickly played back in real time. An image file is created for each frame cached and stored in your project’s Render Pictures folder by default.
You can load the cached images as a flipbook in the standalone flipbook, the native media player, or the image clip viewer—see Previewing Animation in a Flipbook on page 94 for more information.

Capturing the Images

To capture a sequence of images

1. In the viewport whose images you want to capture, set the display options as desired.

2. In the viewport you want to capture, click the camera icon and choose Start Capture from the menu.

   You can also enter the CaptureViewport command in the script editor to start a capture automatically without opening the Capture Viewport dialog box. See the online help in the script editor for the syntax and options for this command.

3. In the Capture Viewport dialog box, set the options as you like.

   For a description of each option, click the Help button in the dialog box.

4. When you click OK, the images in the viewport are cached frame by frame. While XSI is capturing the images, the frame rate is slow; however, subsequent replays will be in real time.

   Click the Stop button in the caching window if you want to end the caching.

Launching a Player after Capturing

To play back the flipbook immediately, select one or both of these options in the Capture Viewport dialog box:

To launch the standalone flipbook

· If you select Launch Flipbook When Done, the standalone flipbook opens and plays back the images when the capturing is done (see Previewing Animation in a Flipbook on page 94).

   If you start the flipbook immediately after capturing the images, the frame rate set in the Playback Options property editor is used.

To launch your computer’s native player

· If you also select Play Videos with Native Players, the appropriate media player immediately opens and plays back the images when the capturing is done. The media player launched depends on your computer’s settings.

   For example, if you select AVI as the file format and the Windows Media Player is the default player for AVI files on your computer, then this player automatically opens when the capture is finished.

   If you’re working on Linux, the default native player is mplayer. To change this default to another player, you have to set the XSI_PLAYER environment variable. If you want to set permanently, add it to your .xsi_4.0 file (in the XSI home directory).
Native players are recommended for playing video files because they take less memory and usually take advantage of hardware (such as DirectDraw on Windows) and other options.

To load the cached images later

- If you didn’t choose to play the images immediately after capturing, you can load the cached images as a flipbook in either the standalone flipbook (as described in Previewing Animation in a Flipbook on page 94) or the image clip viewer (choose Image Clip Viewer from a viewport’s Views menu).

Capturing Animation with Sound

If you want to have sound along with animation, you can do so while you capture in a viewport.

1. Enable the audio by clicking the Play/Mute Audio button on the Playback panel. Audio is enabled when the button doesn’t appear to be highlighted.
2. Capture the animation in the viewport by choosing Start Capture under the Camera icon.
3. In the Capture Viewport property editor, select Record Audio Track.

You can also export a flipbook with the audio file attached—see Exporting a Flipbook on page 98 for more information.
Previewing Animation in a Flipbook

After you have captured animation, you can view it as a flipbook. Unlike animation played back using the Playback panel, a flipbook always plays back at the defined frame rate, provided you have enough memory for the cached images.

There are three ways to view a flipbook:

- Using the standalone flipbook—see this section.
- Using the media player native to the file format you choose.
- Using the Image Clip Viewer—choose Image Clip Viewer from a viewport’s Views menu.

You can export flipbooks in a variety of standard formats, such as AVI and QuickTime (see Exporting a Flipbook on page 98).

Loading Animation in the Flipbook

You can use the standalone flipbook to view cached images as a flipbook. The standalone flipbook is useful for viewing high-resolution images that use a lot of RAM. You can launch the standalone (flip.exe) from the Softimage Products > XSI program menu, a command prompt window or Linux shell, or from within the XSI interface.

To launch the standalone flipbook from outside XSI

Do either of the following:

- Choose Flipbook from the Start > Softimage Products > XSI program menu.
or

1. In a command prompt window or Linux shell, go to the XSI program’s bin directory.
2. Type `flip` at the command prompt to open the flipbook window.
3. Choose Help > Command Line Options to see the syntax and options available.
   You can also type `flip -h` at the command prompt to display the list of the options available to run the flipbook from the command line.

To launch the standalone flipbook from within XSI

1. Choose Playback > Flipbook from the Playback menu at the bottom of the XSI window.
2. In the Open Images dialog box that appears, set the options as you like.
   For a description of each option, click the Help button in the dialog box.
3. Click OK to load the image sequence and open the flipbook window.
   The flipbook window is resized if the loaded image is larger than the current size.

   - If you’re rendering a sequence, you can quickly check it in the flipbook by clicking the Launch Flipbook button on the Output page of the Render Options property editor (choose Render > Render Options from the Render toolbar).
   - You can open the Open Images dialog box later by choosing File > Open Images or pressing Ctrl+o in the flipbook.
     If you’re constantly making changes and testing the same image sequence, leave the flipbook open and press Ctrl+o and then Enter to reload the sequence. This way, all the settings in the dialog box stay the same.

Creating a Shortcut for the Flipbook

To set up a shortcut for the flipbook

1. Copy the file `xsi.bat` in the XSI\Application\bin folder (or create a new batch file) and call it `flip.bat`.
2. Edit the `flip.bat` file and replace `xsi.exe` in the second line to `flip.exe` so that the file looks something like this:

   ```
   @echo off
   call "C:\Softimage\XSI\Application\bin\setenv.bat"
   start "" "C:\Softimage\XSI\Application\bin\flip.exe" "%1" "%2" "%3" "%4" "%5" "%6" "%7" "%8" "%9"
   ```
3. Drag the `flip.bat` file onto your desktop as a shortcut.

**Setting Up the Flipbook View**

In the flipbook window, you can set up how you want the images to appear using commands from its View menu:

- Display different RGB or alpha channels using the commands in the View > Channel menu. Press the `r, g, b, a,` and `c` (all channels) keys to display the respective channels.
- Set the zoom factors using the options in the View > Zoom menu.
- You can have non-square pixels appear correctly on the computer monitor, which has square pixels by choosing View > Correct Aspect Ratio. For example, NTSC images have a pixel ratio of 0.9, meaning the pixels are not square.

To set the flipbook's background color

- Choose View > Background Color to open the color property editor in which you can set the color.

To set the flipbook's display options

- Choose View > Display Options to open the Display Options dialog box, from which you can set the following options:
  - **Image Pixel Ratio** overrides the flipbook’s pixel ratio for the current image.
  - **OpenEXR/HDR Settings > Exposure**: if the image format is `.hdr` or `.exr`, this setting offsets the exposure to brighten or darken the image.
    
    The exposure value is measured in f-stops, so a value of 0 is the image’s default exposure, while non-zero values offset the default by the specified number of f-stops in either direction.
  - **OpenEXR/HDR Settings > Display Gamma**: if the image format is `.hdr` or `.exr`, this setting performs a gamma correction for display purposes only.
Playing the Flipbook

After you’ve loaded the images and set up the flipbook, you can play back the images using the playback controls, as well as set up and adjust the playback rate.

When you play back the flipbook, the current frame number is shown below the image, at the right edge of the window.

To set the playback rate

- Choose Tools > Rate (NTSC, PAL, FILM, etc.) or press Ctrl+r.
- You can also choose View > Play Real-time or press t to display the images in realtime.

To adjust the playback rate

You can adjust the playback rate by 1 fps (frame per second) increments within the range of 24 and 30 fps.

- Choose Tools > Increase Rate or press + (plus) on the number pad to increase one frame per second.
- Choose Tools > Decrease Rate or press - (minus) on the number pad to decrease one frame per second.

To play the animation

These controls are the same as in the Playback panel in the main XSI window, as described in Using the Playback Controls on page 80.

- Click the play icon on the flipbook’s playback controls or press the up arrow key.
You can also move forward or backward frame by frame, go to the first or last frame, stop, and loop the playback (loop is automatically selected).

**To play the animation in ping-pong style**
1. Choose View > Ping Pong or press p. This changes the directions of playback at the ends of a looped sequence.
2. Click the loop icon in the flipbook’s playback controls.
3. Play the sequence: at the first and last frame, the sequence changes directions.
4. To stop the playback, press the space bar.

**To pan in the flipbook**
- Click+drag across the view area. This pans the image when it is bigger than the display area.

**To go to a particular frame**
- Choose Tools > Go to Frame or press Ctrl+g in the flipbook window.

**To change the playback range**
- Use the range control below the timeline:
  - Click+drag either end to change the start/end frames.
  - Click+drag in its middle to slide the time range.
  - Double-click in it to reset it back to the entire range.

**To clear the flipbook window of the current cached images**
- Choose File > Clear Flipbook.

**To close the flipbook window**
- Choose File > Exit or press Alt+F4.

**Exporting a Flipbook**
You can export a flipbook using a number of file formats, as shown below on the left.

Before you export a flipbook, make sure that you have enough room on the disk to where you’re exporting. A warning message will not be displayed if you run out of space.

**To export a flipbook**
1. In the flipbook window, load the sequence you want to export.
2. Choose File > Export in the flipbook’s command bar or press Ctrl+e.

3. In the Export dialog box, specify the File name you want to export and also the Source Range of frames to export.

4. Select the File Type in which you want to export the file. The list here left shows the possible export formats supported.

5. Set these optional parameters:
   - If you have a specific compressor (coder-decoder) you want to use for movie files, click the Codec button.
   - To export an accompanying Audio File with the images, click the (…) button to open the browser and search for the audio file.
   - If you want to pad the numbers to allow you to match up image sequences, specify the number of frames in the Number Padding list. The default syntax is [fn].#[ext] where fn is the base file name, ext is the file format extension, and # is the frame number.

6. In the Destination Frame Offset text box, specify the number of frames to add to the destination images. For example, you can offset the image sequence before using another tool.

7. Click OK.
Chapter 4  Animating with Keys

Keyframing (or “keying”) is a process for animating values over time. Traditional hand-drawn animation is generally created using keyframes—an animator draws the extreme (or critical) poses at the appropriate frames, creating “snapshots” of movement at specific moments.

Keyframes are the cornerstone of convincing animation. The drawings themselves determine what a figure looks like at specific moments in time, the frames chosen for the keyframes determine the timing and weight of a figure, and the number of keyframes can reflect the complexity of the movement. The frames in between the keyframes are added in afterward. This filling-in process is called in-betweening.

As in traditional animation, a keyframe in XSI is also a “snapshot” of one or more values at a given frame, but unlike traditional animation, XSI handles the in-betweening for you, computing the intermediate values between keyframes by interpolation.

Keys set at frames 1, 50, and 100. Intermediate frames are interpolated automatically.
You can set keys for just about anything in XSI that has a value: this includes an object's geometry, colors, textures, lighting, and visibility.

When you set keyframes to animate a parameter, a function curve is created. A function curve (or fcurve) is a graph that represents the animation of a parameter. When you edit a function curve, you change the animation.

Viewing Keys

Viewing and Editing Keys in the Timeline

When you select an object animated with keys, you can see its keys in the timeline. As well, you can edit the keys by creating regions of keyframes, as you do in the dopesheet, and then move, scale, cut, or copy and paste them.

For more information, see Editing Keys in the Timeline on page 125.

Ghosting Keyframes

Ghosting is a way of tracing or drawing ghosted images of an object as it moves from frame to frame. Ghosting works for any animated object.

Drawing ghosts only on keyframes lets you quickly see where an object’s keys are (local and global transformation parameters only). If you add or remove a key with ghosting on, the ghosts are updated to reflect the change.

For more information, see Ghosting Animated Objects on page 42.
Setting Keys

You can set keys for any animatable parameter in any order and at any time. When you add a new key, XSI recalculates the interpolation between the previous and next keys. If you set a key for a parameter at a frame that already has a key set for that parameter, the new key overwrites the old one.

Before you start setting keys, you need to set a preference that can determine the way in which you key: with marked parameters or with keyable parameters. The differences are described in Setting Preferences for Keying on page 105.

While there are different methods for setting keys, this overview shows you one basic workflow to follow. The different methods of setting and deleting keys are described in Methods for Setting and Removing Keys on page 106.
Overview of Setting Keys

1. Select the object you want to animate and go to the frame at which you want to set a key.

2. Set one or more parameter values for the selected object.

3. Mark the parameters you want to key. Transformation parameters are automatically marked when you select them.

4. Set a key for the marked parameters using any method as described in Methods for Setting and Removing Keys on page 106.

5. Go to the next frame at which you want to set a key.

6. Change the values for the marked parameters and set another key.

You can also animate the ball's translation parameters at the same frames to create a basic bouncing ball.

You cannot set keys when you have a render region open and are using motion blur. You must close the render region before keying.

When you set keyframes for a parameter, a function curve is created for that parameter. A function curve (or fcurve) is a graph that represents the parameter’s animation. When you edit a function curve, you change the animation.

You can display and edit the resulting animation in different ways:

- Edit the keys in the timeline—see Editing Keys in the Timeline on page 125.
- Edit the fcurve in the animation editor—see Editing Function Curves on page 159.
Setting Keys

- Edit the keys in the dopesheet, which is similar to a traditional cel animator’s dopesheet—see Editing Animation in the Dopesheet on page 241.

To set up default options for the function curves that are created when you set keys, choose File > Preferences from the main menu and then select Animation or click its icon.

**Setting Preferences for Keying**

Before you start setting keys, you need to choose a preference that can determine the way in which you work: with marked parameters or with keyable parameters.

**To set a keying preference**

1. Choose File > Preferences from the main menu and select Animation, or open the Animation menu in the Animation panel at the bottom of the XSI window.

1. Select the appropriate Save Key Command Mapping option:

   - Choose **On marked parameters** if you want to first mark parameters, then set keys on them. For more information on this, see Marking Parameters for Animation on page 35.

   - This is the default method for working in XSI.

   - Choose **On all keyable parameters** if you want to set keys on all parameters that are displayed in the keying panel (see Keying Parameters in the Keying Panel on page 109). This is an easy way to quickly key all parameters you’ve set up in the keying panel, making the keying panel like a marking set for the object. If you’re using the QWERTY keyboard mapping, this option is on by default.

   - Choose **On marked keyable parameters** to set keys on only keyable parameters that you have marked in the keying panel.

2. Set a key using the methods described in the next section, Methods for Setting and Removing Keys.
Chapter 4 • Animating with Keys

Methods for Setting and Removing Keys

There are a number of ways in which you can set keys in XSI, depending on what you want to key and where. Here’s a quick overview of some of the more common methods—see the rest of this section for more information on each method:

Once you mark parameters, you can set keys on or remove keys from them in any of these ways ...

- Click the keyframe icon. When you’re on a keyframe, clicking this icon removes the key.
- Choose Set Key or Remove Key from the Animation menu.
- Click the auto button to automatically set a key each time you change parameter values.
- Press k to set a key or press Shift+k to remove a key.
- In an explorer, right-click a parameter’s animation icon and choose Set Key or Remove Key.

Use the keying panel to set keys on keyable parameters.

To delete keys, go to a frame where there is a key and use any methods described in this section.

If you want to set keys on fractional frames (between frames), add keys or move them on a function curve. You can set them at any frame or value as you like.

To set keys in the keying panel

- See Keying Parameters in the Keying Panel on page 109.
To set keys for marked parameters

- Go to the frame where you want a key, mark the parameters, and do one of the following:
  
  - Press the k key.
    
    If you press Shift+k when you're at a keyframe, it removes all keys on marked parameters.
  
  or

  - Click the keyframe icon in the Animation panel (in the lower-right corner of the interface). The keyframe icon is red when there's a key at the current frame (see The Keyframe Icon's Color on page 118).

    ![Keyframe icon](image)

    If you click the keyframe icon when you're at a keyframe, it removes keys on all marked parameters.

  or

  - Choose Animation > Set Key in the Animation panel.

To set keys using marking sets

- See Remembering Marked Parameters with Marking Sets on page 39.

To set keys in the explorer

- Right-click a parameter’s animation icon and choose Set Key.

- To remove a key, right-click and choose Remove Key.

  ![Animation menu](image)

  To see the parameter’s value in the explorer, choose Show > Parameter Values in the explorer’s command bar. Then use the virtual slider (press F4 and “scrub” the slider) to change the values as you're setting keys.

  For more information on the slider, see Entering Values Using Virtual Sliders on page 57 in the Interface and Tools guide.

To set keys for parameters in a property editor

- See Keying Parameters in Property Editors on page 117.

To set keys for transformation parameters

- See Keying Transformations on page 146.
To set keys automatically

- If you are keying a great deal, you can automatically set keys for individual parameters as you modify their values—see Setting Keys Automatically on page 121.

Setting Keys at Multiple Frames

To set multiple keyframes at a time

1. Mark the parameters you want to key and choose Animation > Set Keys at Multiple Frames.

2. In the dialog box that opens, enter the frame numbers at which to set keys, separated by commas.

3. When you click OK, keys are set at these frames (based on current values and current marking).

This is a good way to create “place holders” for future keys. If you use this command while autokeying with the On every value change option selected, you can work very quickly (see Setting Up for Autokeying on page 121).
Keying Parameters in the Keying Panel

Using the keying panel, you can quickly and easily change values and set keys for specific parameters of a selected object. The parameters that are displayed in the keying panel are called keyable parameters. You can set which parameters are keyable or not using the keyable parameters editor.

You can view parameters in the keying panel for only one selected object at a time. If you have multiple selected objects, you can select which one's parameters to display in the keying panel using the keyable parameters editor.

The keying panel lets you work efficiently because you display only the parameters you need for keying. If you are setting keys on keyable parameters, you can use the keying panel like a marking set because all parameters in it are keyed at once. Or you can mark parameters in the keying panel, as you may be used to in XSI, and then set keys on only those parameters.

Here's a quick overview of the steps you can do to set keys using the keying panel. See the following page for more details on each step.

Overview of Setting Keys in the Keying Panel

1. Set the preference for saving keys to keyable parameters.
2. Select an object and open the keying panel.
3. If you need to add other keyable parameters to the keying panel, select them in the keyable parameters editor.
4. Go to a frame where you want to set a key.
5. Change the values for the keyable parameters.
6. Set a key for these parameters at this frame.
To set keys in the keying panel for the selected object

1. Set the appropriate Save Key Command Mapping option from the Animation menu in the Animation panel, or by choosing File > Preferences from the main menu and selecting Animation:

   - Select On all keyable parameters if you want to set keys on all parameters displayed in the keying panel. This is an easy way to quickly key all parameters you've set up in the keying panel, allowing you to use the keying panel as a marking set for the object.
   
   If you're using the QWERTY interaction model, this option is on by default.

   or

   - Select On marked parameters or On marked keyable parameters if you want to first mark parameters, then set keys on them. This is a keying method that is the default XSI way.
   
   For general information on marking, see Marking Parameters for Animation on page 35.

2. Select an object and do either of the following to open the keying panel:

   - Click the KP/L tab at the bottom of the main command panel to anchor the keying panel in the right panel of the XSI window.

   or

   - Choose View > Animation > Keying Panel from the main menu to open the keying panel in a floating window.

   By default, the selected object's local transform parameters and visibility parameters are displayed.

   You can also display an object’s proxy parameter sets and marking sets by setting a preference in the Keying Panel preferences property editor: choose Parameters > Preferences from the keying panel and select Show Proxy/Marking Psets.

3. If you need to, open the keyable parameters editor in which you can select which parameters to display and key in the keying panel. These parameters are considered to be keyable parameters (see Choosing Which Parameters to Display in the Keying Panel on page 111).

4. Go to a frame where you want to set a key.

5. Set the values for the parameters in the keying panel (see Changing Parameter Values in the Keying Panel on page 113).

   If you are working with marked parameters, you need to first mark the parameters you want to animate, then set their values.
6. Set a key for the keyable parameters in the keying panel (see Setting Keys in the Keying Panel on page 114).

If you are working with marked parameters, only the marked parameters are keyed.

Choosing Which Parameters to Display in the Keying Panel

When you open the keying panel, the selected object’s Local Transform and Visibility parameters are displayed by default. However, if you want to display and key other parameters in the keying panel, you need to make them keyable using the keyable parameters editor.

Keyable parameters are simply parameters that you have flagged as being available to be displayed and keyed in the keying panel.

Making Parameters Keyable

1. Do either of the following to open the keyable parameters editor:
   - Choose Parameters > Keyable Parameters Editor in the keying panel.
   or
   - Choose View > Animation > Keyable Parameters Editor from the main menu.

2. If you have multiple objects selected, select which one you want to modify from the object list in the keyable parameters editor.

3. Select a parameter filter type from the Parameters menu in the keyable parameters editor to determine which type of parameter is displayed in the parameter explorer:
Chapter 4 • Animating with Keys

- **All Nodes** displays all parameters of the selected object.
- **Local Transform** displays only the parameters in the local transformation node (basically, position, rotation, and scaling) of the selected object.
- **Visibility** displays only the parameters in the visibility node of the selected object.
- **Custom** displays only custom parameter sets for the selected object. These include marking sets, which are a type of custom parameter set.

4. In the parameter explorer, select one or more parameters that you want to make keyable.

5. Click the **Add** button below this explorer.

   This parameter is displayed in the keyable parameter list (on the right) and is immediately displayed in the keying panel.

**Making Parameters Non-Keyable**

You can make parameters non-keyable in either the keyable parameters editor or in the keying panel itself.

To make parameters non-keyable in the keyable parameters editor

Do either of the following:

- Clear the checkbox of a keyable parameter from the list on the right side of the keyable parameters editor. This makes the parameter non-keyable, but still visible in the keying panel. This type of parameter is displayed in green in the keying panel.

  or

- Select a keyable parameter from the list and click the **Remove** button below this list. This parameter moves to the explorer on the left and is immediately removed from the keying panel.

To make parameters non-keyable in the keying panel

- Mark a parameter in the keying panel, then right-click and choose either of these commands:
Keying Parameters in the Keying Panel

- **Make Non-Keyable and Hide** makes the parameter non-keyable and is removed from the keying panel.

  or

- **Make Non-Keyable** makes the parameter non-keyable, but it remains visible in the keying panel and is displayed in green.

**To apply keyable state changes to all objects in the list**

- If you have multiple objects in the object list, shown above the parameter explorer, you can select the **Change for all Objects in List** option to apply the same type of changes you make to the keyable parameters to all objects in the list.

  For example, if you select this option and then remove the Visibility parameters from the keyable parameter list or clear their checkboxes, the same is done for the corresponding parameters for all objects in the list.

  You can set a preference for the state of this option (on or off) in the Keying Panel preferences property editor: choose **Parameters > Preferences** from the keying panel.

**Changing Parameter Values in the Keying Panel**

**To change individual parameter values**

- Scrub in the field beside the parameter name by clicking and dragging the mouse to the left or right.

  or

- Click in the field so that it’s highlighted, then enter a value.

**To change multiple parameter values**

1. Drag across the fields to select multiple adjacent fields.

  or

   - Press Ctrl+click to select random multiple fields.

     A marquee box appears around the selected fields.

2. Enter a value in the active field (either the top or bottom field depending on which direction you dragged), then press Enter.

   Press the Tab key to go from one field to the next for quick editing.
All selected fields have the value that you entered.

Setting Keys in the Keying Panel

The keying panel lets you work efficiently because you display only the parameters you need for keying. Depending on how you like to work, you can choose to set keys on either keyable or marked parameters in the keying panel:

- Setting keys on keyable parameters lets you use the keying panel like a marking set because all parameters in it are keyed at once.
- Setting keys on marked parameters is a method you may be used to from working in XSI.

To set keys for keyable parameters in the keying panel

1. Select the On all keyable parameters option for the Save Key Command Mapping command in the Animation menu in the Animation panel.
   
   If you’re using the QWERTY keyboard mapping, this option is on by default.

2. Go to a frame where you want to set a key on the keyable parameters and do any of the following:
   
   - Press the k key (or press s if you’re using the QWERTY keyboard mapping). Press Alt+k to key all keyable parameters in the branch.
     
     or

   - Click the key icon at the top of the keying panel or in the Animation panel at the bottom of the XSI window.
     
     If you click the key icon when you’re at a keyframe (the icon is red), it removes keys on all parameters.
     
     or

   - Choose Parameters > Key in the keying panel.

   Choose Parameters > Remove Key to remove keys for the parameters in the keying panel.
Keying Parameters in the Keying Panel

To set keys for marked parameters in the keying panel

1. Select the On marked parameters or On marked keyable parameters options for the Save Key Command Mapping command in the Animation menu in the Animation panel.

2. Mark the parameters in the keying panel.

   Mark parameters by clicking on their name.

   Press Shift or Ctrl+click to mark multiple parameters.

3. Go to a frame where you want to set a key on the marked parameters and do any of the following:
   - Press the k key.
   - Click the key icon at the top of the keying panel or in the Animation panel at the bottom of the XSI window.
     If you click the key icon when you're at a keyframe (the icon is red), it removes keys on all marked parameters.
   - Choose Parameters > Key in the keying panel.
     Choose Parameters > Remove Key to remove keys for the marked parameters in the keying panel.

To set a key for all parameters in the keying panel

- Choose Parameters > Key All in the keying panel.
  Choose Parameters > Remove Key All to remove keys for all parameters in the keying panel at this frame.

To set a key for individual parameters

- Click the parameter's animation icon.
  - Right-click on a parameter's animation icon and choose Set Key.
Muting Animation in the Keying Panel

To mute the animation for parameters

- Mark the parameters whose animation you want to mute and choose Parameters > Deactivate Animation. The parameter’s animation icon turns white when muted.
  
or
- Mark parameters, then right-click and choose Deactivate Animation.

To unmute the animation for parameters

- Mark the parameters whose animation you want to reactivate (unmute), then choose Parameters > Activate Animation.
  
or
- Mark parameters, then right-click and choose Activate Animation.

To mute/unmute the animation for all parameters in the keying panel

- Choose Parameters > Deactivate All Animation. The parameters’ animation icons turn white when muted.
  
- Choose Parameters > Activate All Animation to reactivate the animation for all parameters in the keying panel.
Keying Parameters in Property Editors

Every property editor lets you set keys for any or all of its animatable parameters. An animatable parameter is identified by an animation icon (little green box) beside it. You can set keys in two ways:

- Click the keyframe icon at the top of the property editor to key all or only the marked parameters on the current property page.

or

- Click a parameter’s animation icon to key only that parameter.

Keying All or Marked Parameters

You can either mark parameters in the property editor for keying or key all parameters in the property editor.

To key all or marked parameters on a property page

1. Move the playback cursor on the timeline to a frame where you want a key and open the property editor.

2. Display a property page with parameters you want to animate.

3. Mark the parameters you want to animate (click their names so they turn yellow) and set their values.

or

If nothing is marked and you have selected the Apply to All Parameters When Nothing Is Marked option (as described next), all parameters on the page are keyed.

4. Click the keyframe icon at the top of the property editor.

This sets keys (or removes a key if there is already one at the current frame) for the corresponding parameters on the property page.
5. Move the playback cursor to another frame where you want a key.
6. Set the values of the marked parameters and click the keyframe icon again.

**To key all parameters on a property page**

1. Choose File > Preferences from the main menu.
2. In the explorer that appears, select Animation to open the Animation Preferences property editor.
3. Make sure that Apply to All Parameters When Nothing Is Marked is selected in the Animation Scope in Editors area.

Then when you click the keyframe icon, it saves keys for all the parameters on that page at their current values.

**The Keyframe Icon's Color**

The keyframe icon's color changes, depending on the status of the property page's marked parameters.

- Gray: no marked parameters have been animated.
- Red: a key is defined for a marked parameter on the current property editor page.
- Green: a marked parameter is animated but the current frame is not a keyframe.
- Yellow: an animated marked parameter on the current property page has been modified but is not yet keyed.

If a property editor’s marked parameters have different animation icons, the keyframe icon's color is based on priority:

- It is red only if there are no yellow animation icons.
- It is green only if there are no yellow or red animation icons.

**Keying Specific Parameters**

1. Open the property editor and display the property page with the parameters you want to animate.
2. Move the playback cursor to a frame where you want a key.
3. Set the value for a parameter you want to animate and click the animation icon to the left of its name, or right-click it and choose Set Key. Doing this sets keys for the parameter whether it is marked or not.

Repeat steps 2 and 3 for all the parameters you want to animate in the property editor.

If you want to key parameters in other property pages or property editors at the current frame, display them and repeat these steps.
Moving between Keys in a Property Editor

There are several ways to move between keys defined for parameters on the currently displayed property page.

- If there are marked parameters on the property page, the commands move between the keys for only those parameters.
- If nothing is marked, the commands move between keys for all parameters on the property page.

To move between keys for parameters on a property page

- Press the , (comma) key to select the previous keys, and press the . (period) key to select the next keys.

  or

- Click the previous or next arrow icons in the upper-left corner of the property editor.

  or

- Right-click the marked parameter’s animation icon and choose Previous Key or Next Key.

  or

- To go to the first or last keys defined, right-click the keyframe icon and choose First Key or Last Key. The keyframe icon changes color to help you navigate among keyframes—see The Keyframe Icon’s Color on page 118 for more information.

Removing Keys in a Property Editor

To remove all keys from all parameters on a page at a specific frame

1. Display the property editor and the specific page from which you want to remove animation.
2. Move the playback cursor on the timeline to a frame where you want to remove the keys.
3. Click the keyframe icon in the property editor so that it is no longer red.

To remove all keys from marked parameters

1. Display the property editor from which you want to remove animation.
2. Mark the parameters whose keys you want to delete.
3. Right-click the keyframe icon and choose Remove Animation.

To remove a single key from a parameter on a property page

1. Open the property editor and the specific page from which you want to remove a key.
2. Move the playback cursor to the keyframe you want to remove.
3. Click the parameter’s animation icon, or right-click it and choose Remove Key.
Setting Keys Automatically

You can have XSI automatically set a key each time you move to a new frame and modify the values of specific parameters.

As well, when you play back the animation with autokey on and use the sliders to set values in a property page, the keys get set for the current time. This may be useful for roughing out an initial animation.

If you go back to the first frame and click the Play icon again after setting keys, keys are only set when you make changes to the slider. This means that if you have animation already there, it won’t be automatically “merged” with the changes you are keying.

Setting Up for Autokeying

If you frequently animate objects and properties, you can automatically set keys as you adjust the object’s properties. Instead of clicking the keyframe icon in a property editor each time you want to add a key, you can let the autokey mode automatically record the adjustments that you make.

This has the same effect as activating the auto button in the Animation panel. The difference between the two is that setting the default mode is saved to your preferences file.

To activate autokey as the default

1. Choose File > Preferences from the main menu.
2. In the explorer that appears, select Animation to open the Animation Preferences property editor.
3. Select the Autokey > Enable Autokey option and then select any of these options:
   - On Any Value Change sets a key whenever you change any properties. When you autokey with this option, the auto button turns red.
   - On Existing Keys Only generates a new key when a parameter is changed only if there’s already a key for that parameter at the current frame. This makes it easy and safe for editing existing keys, allowing you to work in a mode similar to traditional animation. When you autokey with this option, the auto button turns light orange.
   - On Existing Fcurves Only generates new keys when a parameter is changed only if the parameter already has an fcurve. New keys are added regardless of whether or not there’s already a key at the current frame. When you autokey with this option, the auto button turns dark orange.

When you save this setting, the auto button in the lower-right corner of the Animation panel is activated and changes the appropriate color.
Autokeying

To set keys automatically

1. Move the playback cursor to a frame where you want a keyframe (where you want the animation to start).
2. Open the property editor and set the properties to the values you want or transform the objects as you want.
3. Click the auto button in the property editor or the Animation panel.
4. Move the playback cursor to where you want the next keyframe and change the animated properties again. Every time you change a property value, it is keyed.
   Repeat this step as often as you need.
5. When you have finished, click the auto button again to turn autokey mode off.

Be careful when dollying, orbiting, or framing in viewports while autokey mode is on. If you are looking through a camera or spotlight viewpoint, the camera or light is moved and keyed.

Remember that selecting any transformation tool automatically marks the corresponding scale, rotate, or translate parameter. Therefore, using a transformation tool after activating auto sets a key for the transformation.
Moving Between Keys

There are a number of ways in which you can move between keys for all parameters of a selected object, marked or not.

- If the selected object has marked parameters, the commands move between the keys for only those parameters.
- If nothing is marked, the commands move between keys for all parameters of the selected object.

To move between keys of a selected object’s parameters
1. Select or branch-select an object.
2. Do one of the following:
   - Press Ctrl+left arrow or Ctrl+right arrow for previous or next keys, respectively.
   - Click the previous or next arrow icons in the Animation panel.
   - Choose Animation > Previous Key or Next Key in the Animation panel.

To move to the first or last keys of a selected object’s parameters
1. Select or branch-select an object.
2. Do one of the following:
   - Press Ctrl+up-arrow or Ctrl+down-arrow for first or last keys, respectively.
   - Choose Animation > First Key or Last Key in the Animation panel.
Chapter 5

Editing Keys in the Timeline

You can view and edit keys in the timeline similar to how you do in the dopesheet. The advantage of doing this in the timeline, of course, is that you don't need to open up a separate editor: the keys are right there. This lets you keep the object that you're animating in full view at all times.

Once you have selected an animated object, you can easily move its keys, cut or copy and paste its keys, and scale a region of frames with keys, all within the timeline. This is especially useful for blocking out rough animations before you do more detailed editing.
You can also display and edit keys on function curves in the animation editor (see *Editing Function Curves* on page 159) and display and edit keys in the dopesheet (see *Editing Animation in the Dopesheet* on page 241).

**Viewing Keys in the Timeline**

*To view keys in the timeline*

- Select an object with keyframes set on any of its parameters.

The keys appear as red lines in the timeline. The thickness of the lines depends on the time range: the fewer the number of frames in the timeline, the thicker the lines appear.

*To choose which type of keys are displayed in the timeline*

- Right-click in the timeline and choose an option from the *Show Keys* menu:
  - *None* displays no keys.
  - *All* displays keys for any type of parameter. For example, this is how you can display shape keys.
  - *Marked Parameters* displays keys only for marked parameters.
  - *Keyable Parameters* displays keys only for parameters that are displayed and set as keyable in the keying panel (see *Keying Parameters in the Keying Panel* on page 109).
  - *Marked and Keyable Parameters* displays keys only for parameters that are both marked and set as keyable in the keying panel.
  - *Kinematics* displays keys only for the local position, rotation, and scaling parameters.
  - *Position/Rotation/Scaling* displays keys only for each of these type of parameters.
  - *Colors* displays keys only for the individual color parameters. The parameters displayed depends on the color model you have chosen: RGB, HLS, or HSV.
  - *Activation Regions* displays keys only for the Active or Mute parameters.

**Editing Keys in the Timeline**

In the timeline, you can edit your animation by defining regions of keys and moving (see *Moving Keys in the Timeline* on page 129), scaling (see *Scaling Keys in the Timeline* on page 131), cutting, copying and pasting these regions (see *Cutting, Copying, and Pasting Keys in the Timeline* on page 133). You can also select a single key and move, cut, copy, and paste it.
To draw a region

- Press Shift and drag in the timeline in either direction over the frames you want to edit.

The region is displayed as a light gray rectangle on the track with the start and end frame numbers displayed in white.

To deselect the region

- Draw another region or click anywhere else in the XSI interface.

To select a single key

- Press Shift and click on the keyframe.

Rippling

You can use the Ripple command to control whether keys are pushed along the timeline or not when you move, scale, or cut, copy, and paste regions.

When Ripple is on and you perform any of these operations, the keys are pushed to the right (forward in time) or left (backward in time) in the timeline. Any offset between the region and the other keys is preserved. For example, if a region of 10 frames is scaled to 15 frames to the right, the keys on the right are pushed in that direction by 5 frames.

As well, keys automatically snap to frame boundaries: you cannot have a key between frames.

When you ripple keys, you may push them out of the timeline’s range. To help you see all frames available in the scene, display the time range below the timeline by choosing View > Optional Panels > Time Range Slider from the main menu.
Chapter 5 • Editing Keys in the Timeline

To activate Ripple

- Right-click in the timeline and choose Ripple from the menu so that a check mark appears beside it.

You can set the default state for the Ripple command in the timeline dopesheet preferences (see Setting Preferences for Keys in the Timeline on page 138).

Viewing Audio Waveforms in the Timeline

If the selected object has one or more audio clips loaded in its animation mixer (see Audio and Animation on page 199 in the Nonlinear Animation guide), you can display waveforms for them in the timeline. This helps you synchronize the animation with the sound because you can easily see where each audio clip occurs in time.

To view an audio waveform in the timeline

1. Make sure that there is an audio clip on a track in the model’s animation mixer.

2. Right-click in the timeline and select an audio file from the Audio Clip menu. The available audio clips are listed in alphabetical order.

   The audio waveform is displayed in light gray behind the frames and keys.
Moving Keys in the Timeline

When you move keys in the timeline, you are changing the time reference of the animation.

To move a single key in the timeline
1. Press Shift and click once on a keyframe.
2. Middle-click and drag the key to a new frame. The key snaps to the closest frame.
   If you left-click and drag the key, you can shift the region along the timeline, but not the key.

To move a region of keys
1. Make sure that Ripple is on or off, depending on what you want to do (see below).
2. Press Shift and drag in the timeline to create a region of frames including keys.
3. Middle-click in the region and drag its region to the left or right, then release the mouse button at the frame at which you want the keys. The keys snap to the closest frame when you release the mouse button.
   If you left-click and drag the region, you can shift the region along the timeline, but not the keys.

To move with or without rippling
You can move the keys with or without moving the other keys on the timeline, depending on whether Ripple is on or off.

• To move the keys and push the existing keys (ripple) to the right (forward in time) or to the left (back in time), right-click in the timeline and turn the Ripple command on.
To move keys without moving other keys in the timeline, right-click in the timeline and turn the **Ripple** command off. If you move keys on top of existing keys in the timeline, the existing keys are overwritten.
Scaling Keys in the Timeline

When you scale a region of keys, you speed up or slow down the animation within it. The keys within the region adjust proportionally to accommodate the new size. Increasing the region’s size slows down the animation by increasing the number of frames between keys, while reducing the region’s size speeds up the animation by decreasing the number of frames between keys.

- You can do overall scaling of animation using the **Sequence** commands in the Animation menu (see *Scaling and Offsetting (Retiming) Animation* on page 55) or using the dopesheet (see *Scaling Regions* on page 255).

- If you want to scale all the animation in a scene, branch-select the scene root. This displays all keys for all animated objects in the timeline which you can then resize using a region.

**To scale a region of keys**

1. Make sure that Ripple is on or off, depending on what you want to do (see below).

2. Right-click in the timeline and activate **Scale and Translate**.

3. Press Shift and drag in the timeline to create a region of frames.

4. Move the cursor to either end of the region so that it changes to a single arrow, then drag to the left or right. The keys snap to the closest frame when you release the mouse button.

**To scale with or without rippling**

You can scale the keys with or without moving the other keys on the timeline, depending on whether Ripple is on or off.

- To scale the region and push other keys to the right (forward in time), right-click in the timeline and turn the **Ripple** command on. If you shrink the region (scale down), the keys outside it are “pulled” toward the region.

Drag the region’s edge to scale it and push keys with **rippling on**.

This region is scaled forward by 6 frames and pushes the other keys ahead by 6 frames, keeping the same frame offset between the region and the other keys.
With Ripple on, you can also scale a region that has no keys in it (an empty region) and have it affect the other keys on the timeline. This is useful for easily shifting keys in long animations.

- To scale a region without changing keys outside the region, right-click in the timeline and turn the Ripple command off. If you scale the region over existing keys in the timeline, the existing keys are overwritten.
Cutting, Copying, and Pasting Keys in the Timeline

You can cut, copy, and paste one or more keys in the timeline to edit an object's animation. Cutting removes the keys from the timeline and puts them into the paste buffer until they are pasted. Copying puts a copy of the keys into a paste buffer, where they stay until they are replaced, allowing you to paste the same keys as many times as you like.

You can paste the cut or copied keys within a region, in which case the pasted animation is scaled up or down to fit into the target region.

Cutting Keys

To cut keys
1. Make sure that Ripple is on or off, depending on what you want to do when pasting (see Pasting Keys on page 134).
2. Activate the Merge Keys or Replace Keys option (see Merging or Replacing Existing Keys When Pasting on page 135), depending on what you want to do when pasting the cut keys.
3. Press Shift and drag in the timeline to create a region of frames.
   or
   Press Shift and click once on a single keyframe.
4. Right-click in the timeline and choose Cut.

Copying Keys

To copy keys
1. Make sure that Ripple is on or off, depending on what you want to do when pasting (see Pasting Keys on page 134).
2. Activate the Merge Keys or Replace Keys option (see Merging or Replacing Existing Keys When Pasting on page 135), depending on what you want to do when pasting the copied keys.
3. Press Shift and drag in the timeline to create a region of frames.
   or
   Press Shift and click once on a single keyframe.
4. Do one of the following to copy the keys:
   - Press Ctrl and drag the key or region to the frames where you want to copy the keys. Release the mouse button to paste the key or region there.
     or
   - Right-click in the timeline and choose Copy.
Pasting Keys

To paste the keys

1. Press Shift and draw a region into which you will paste the cut or copied keys, or move the playback cursor to the frame where you want to paste.

2. Right-click on the target region or frame and choose Paste.
   - If the region into which you are pasting is smaller than the region of cut or copied keys, the keys are squeezed into the new region, resulting in the animation being faster.
   - If the region into which you are pasting is larger than the region of cut or copied keys, the keys are spread out in the new region, resulting in the animation being slower.

3. If you have copied keys, you can continue pasting the clipboard contents to other regions on the timeline.

To paste keys with or without rippling

You can paste the cut or copied keys with or without moving the other keys on the timeline, depending on whether Ripple is on or off.

- To paste the keys and push the existing keys to the right (forward in time), right-click in the timeline and turn the Ripple command on.

Copying and pasting regions without keys (empty regions) with Ripple on is an easy way to offset the animation. The number of frames in the blank space you copy doesn’t matter: it’s the size of the region into which you paste the blank frames that determines the amount that the animation ripples.

- To paste the keys without moving other keys in the timeline, right-click in the timeline and turn the Ripple command off.
Cutting, Copying, and Pasting Keys in the Timeline

If you paste keys on top of existing keys in the timeline, the existing keys are either merged with the pasted keys or are replaced, depending on which option you have chosen (see the next section, Merging or Replacing Existing Keys When Pasting). With either option, keys that are pasted at the same frame as existing keys overwrite the existing keys.

Merging or Replacing Existing Keys When Pasting

When you paste keys, you can choose to either merge the cut or copied keys with the existing keys in the target (destination) region, or you can replace the existing keys with the cut or copied keys.

To merge or replace keys when pasting

- Right-click in the timeline and activate either of these commands:
  - Merge Keys When Pasting. Cut or copied keys are merged with any existing keys in the target region. If there are overlapping keys, the existing keys are overwritten with the cut or copied keys.
    
    ![Region of keys that are copied.](image)

    ![Target region into which the copied keys will be pasted.](image)

    ![Target region after keys are pasted and merged. The copied keys are spread to fit into the larger region. The copied keys overwrite any existing keys that are on the same frame.](image)
  
  or

  - Replace Keys When Pasting. Any existing keys in the target region are first removed, then the cut or copied keys are pasted in.

    ![You can set the default state for these pasting options in the timeline dopesheet preferences (see Setting Preferences for Keys in the Timeline on page 138).](image)

Keying the Current Values at a Different Frame

As you’re setting keys, you can middle-click the playback cursor on the timeline and drag it to a different frame without updating the scene. The cursor and frame number beside it (and in the current frame box) turn green. To refresh the scene, click the cursor.

By middle-clicking+dragging a frame to another frame, you can copy the current values at this frame to any number of other frames and then set keys. This technique can also be used if you’ve made the mistake of changing the animated values before moving the playback cursor to a new key position.

To key current values at a different frame

1. With the animatable values set at the current frame, do either of the following:

   ![Animation • 135](image)
- Middle-click+drag the playback cursor to its new position on the timeline.

or

- Right-click in the timeline and choose Set Key Time.

The scene does not refresh, so your object’s values do not change. The playback cursor and current frame number beside it turn green, indicating that it is waiting for the key to be set. The key time is also shown in green in the current frame box.

2. Set the key using any method as described in Methods for Setting and Removing Keys on page 106, such as pressing k.

Once the key is set, the red playback cursor jumps to the same position as the green playback cursor and a key is created at this frame. The frame number in the current frame box is displayed in white again.
Deleting Keys in the Timeline

Deleting keys removes the keys you have selected in the timeline, either a single key or a region of keys. Deleting does not put the keys in the clipboard buffer as does cutting, but you can undo deleting keys like anything else in XSI by pressing Ctrl+z.

To delete keys from regions
1. Press Shift and drag in the timeline to create a region of keys.
   
   or

   Press Shift and click once on a single keyframe.

2. Right-click on a region and choose **Delete Keys**.

   Don’t press the Delete key! This deletes the selected object, not the keys in the timeline.
Setting Preferences for Keys in the Timeline

There are several preferences you can set for your default setup while working with keys in the timeline.

To set the timeline key preferences

1. Do either of the following:
   - Choose File > Preferences from the main menu.
   - Choose Edit > Preferences in the dopesheet command bar.
2. In the explorer that appears, select Editors > Dopesheet or click its icon.
3. In the Dopesheet Preferences property editor, click the Timeline Dopesheet page.
   - Select Ripple on the Timeline Dopesheet to have Rippling active by default—see Rippling on page 127.
   - Interaction Mode determines whether the Translate and Scale tool (can either move or scale keys) or the Translate tool (can only move keys) is the default tool.
   - Paste Key Model determines whether you merge or replace keys when pasting them in a region—see Merging or Replacing Existing Keys When Pasting on page 135.
   - Display Keys lets you set the default selected parameters for which keys are displayed in the Show Keys menu when you right-click in the timeline—see Viewing Keys in the Timeline on page 126.
Chapter 6  Animating Transformations

Animating the transformations (scaling, rotation, and translation) of objects is something that you will be doing frequently. It is one of the most fundamental kinds of animation in XSI.

This section explains some of the important things to know about animating transformations. However, you should first be familiar with transformations in general: for a full explanation of all issues related to transformations, see Transformations on page 27 in the Transformations guide.

Before you start animating, make sure you're using the correct scaling method for your intended purpose: Softimage (Hierarchical) or Classical. You can set the method as described in Hierarchical (Softimage) Scaling versus Classic Scaling on page 58 in the Transformations guide.
Tools for Animating Transformations

These are a few key tools that you use frequently when animating transformations.

- The Transform panel, located in the main command panel on the right of the XSI window, is where you can scale, rotate, or translate any object in XSI. Each type of transformation has different manipulation modes, the importance of which is described in Manipulation Modes versus Transformation Values on page 143.

- When you transform an object, you can find transformation information in the object's Kinematics node in the explorer. Kinematics in this case refers to “movement,” not inverse or forward kinematics.

Within the Kinematics node are the Global Transform and Local Transform nodes, referring to the type of transformation.

Then within each of the Transform nodes, there are the Pos (position, also called translation), Ori (orientation, also called rotation), and Scl (scale) folders.

Each of the Pos, Ori, and Scl folders contain the X, Y, and Z parameters corresponding to the particular axes.
The **Kinematics property editor** contains the object’s set of transformation controls. The Kinematics property editor shows both the Local Transform and Global Transform properties, and can be used to modify an object’s scaling, rotation, and translation in X, Y, and Z, as well as other options.

You can also open an object’s Global or Local Transform property editor by itself.

Clicking on an object’s Kinematics icon in an explorer displays transformation controls in the Kinematics property editor.

To quickly open only the Local Transform property editor, select an object and press Ctrl+k.
Chapter 6 • Animating Transformations

Animating Local versus Global Transformations

You can animate objects either in terms of their parents (local animation) or in terms of the scene’s world origin (global animation). When you do this, the appropriate local or global transformation values are stored, as you can see in the Local and Global Transform property editors.

Local Transformations

While you can animate either the local or the global transformation values, it’s usually better to animate the local transformations. This is because you usually animate relative to the object’s parent instead of animating relative to the world origin. Animating locally lets you branch-select an object’s parent and move it while all objects in the hierarchy keep their relative positions.

When you activate a tool in the Transform panel, the corresponding local transformation parameters are automatically marked for animation, assuming that you usually want to animate this way. You can then immediately key these values (see Chapter 4: Animating with Keys on page 101) without having to manually specify what you want.

Global Transformations

The global transformation values are the final result of all the local transformations that are propagated down the hierarchy from parent to child.

- If you animate the global transformations, it does not matter if your objects are in a hierarchy or not. Nothing is inherited if you have global transformation keys because they override any parent-to-child inheritance.

- If you want to animate global transformation parameters, you must explicitly mark (and lock) them in the marked parameter list or the Global Transform property editor (see Marking Transformation Parameters on page 144).

- If you always want to manipulate and key an object globally (not have it altered by its parent transformations in any way), you can change how the inheritance works on the Options page in the object’s Local Transform property editor. See Changing the Animation Inheritance on page 148 for information.

If you animate both the local and the global transformations, the global animation takes precedence.
Manipulation Modes versus Transformation Values

When you transform interactively in a 3D view, you do so using one of several modes. These modes determine which coordinate system is used for manipulation. The manipulation mode affects the interaction only, the resulting values of which you see in the text boxes in the Transform panel.

This is important to know, particularly for understanding the Local manipulation mode: the values shown in the Transform panel while using a transformation tool are not necessarily the same as the local transformation values that are stored for the object (the values that you can animate).

The Local manipulation mode refers to transformations of an object relative to its own local axes (a self reference); however, local animation refers to animating transformations of an object relative to its parent. (You cannot animate relative to one’s own axes). If an object has no parent, the stored local value is relative to the scene root (which is represented by the world origin).

The Global manipulation mode and global animation both refer to transformation relative to the origin of the scene's global coordinate system, so there's no confusion here.

So, how do you manipulate an object so that the values on the Transform panel are the same as the stored values for local animation?

- When you translate, use the Par (parent) mode, which controls the object’s translations relative to its parent, as does local animation. This is the only translation manipulation mode where the axes of interaction correspond exactly to the coordinates of the object’s local position for the purpose of animation. When you activate individual axes on the Transform panel, the corresponding local position parameters are automatically marked.

- When you rotate, use the Add (additive) mode, which controls the object’s rotations relative to its parent, as does local animation. This is the only rotation manipulation mode where the axes of interaction correspond exactly to the coordinates of the object’s local orientation for the purpose of animation. When you activate individual axes on the Transform panel, the corresponding local orientation parameters are automatically marked. This mode is especially useful when animating bones and other objects in hierarchies.

Of course, you can set and animate the values as you like directly in either the Local or Global Transform property editor.
Chapter 6 • Animating Transformations

Marking Transformation Parameters

By default, when you activate any of the transformation tools (scale, rotate, or translate) from the Transform panel or using their hot keys, their corresponding local transformation parameters are automatically marked. To mark specific local transformation axes or global transformation parameters, however, there are some things you must do.

Marking Specific Local Transformation Axes

Let’s say you’re rotating only in Y in Local manipulation mode, and now you want to key only the Rotation Y parameter. When you look in the marked parameter list, you notice that all three rotation axes parameters (Ori.Euler) are marked, not just the one for Y. Why is this?

It’s because the local manipulation mode does not refer to the local animation: it refers to the object transforming relative to its own center. Since there are no special parameters to key the effect of animating relative to the object’s centre, the entire set of local transformations for the object is marked.

There are three ways to have the specific X, Y, or Z axes marked for local animation:

• Rotate in Add mode or translate in Par mode. These are the only two manipulation modes that transform in the same way as local animation: they are both relative to the object’s parent.

• Explicitly mark whatever parameter you want in the marked parameter list (see Marking Parameters for Animation on page 35) or Local Transform property page (see Tools for Animating Transformations on page 140), regardless of the manipulation mode.
Marking Transformation Parameters

Marking Global Transformation Parameters

- Choose **Transform > Automark Active Transform Axes** on the Transform panel. Then when you click a transformation’s specific axis button (such as the Rotation’s Y button) on the Transform panel to manipulate an object, only that axis is marked, regardless of the current manipulation mode.

If you want to animate global transformation parameters, there are two ways to do this:

- Explicitly mark (and lock) them in the marked parameter list or the Global Transform property editor.
- Choose **Transform > Automark Global Transforms** on the Transform panel. Then when you manipulate in a Global mode, the global transformation parameters are marked instead of the local ones.

If you also choose **Transform > Display Global Coordinates in View Mode**, the SRT text boxes in the Transform panel display global values when in View Reference mode and the global transformation parameters are marked in both View and Global mode.
Keying Transformations

Keying transformations (scaling, rotation, and translation) is something that you’ll probably do a lot. As such, there are a number of ways in which you can key the transformation of scene elements.

You can plot a curve (trajectory) from an object’s animated transformation values—see Creating a Curve by Plotting an Object’s Transformation on page 274.

To key transformations

1. Select one or more objects.
2. Move the playback cursor to a frame where you want a keyframe.
3. Use any of these methods to set the transformation values:
   - Use the SRT controls in the Transform panel, as shown on the left.
   - Press the shortcut keys for scaling, rotation, or translation (x, c, or v, respectively).
   - Use the 3D transform manipulators (press b to display them).
   - Open the Local Transform (press Ctrl+k) or Global Transform property editors for an object and key the transformation parameters there.
   You can also open the Kinematics property editor, which contains both the Local and Global Transform property editors, by right-clicking Selection in the Select panel and choosing Kinematics.
   - Use the explorer or marked parameter list to mark and key specific transformation parameters.
4. Set a key using any method as described in Setting Keys on page 103, such as pressing the k key or clicking the keyframe icon in the Animation panel.
5. Move the playback cursor in the timeline to another frame where you want a key.
6. Transform the elements and set another key.

Camera rotation in the X and Y axes cannot be keyed because the camera is constrained to point at the camera interest. You can animate a camera’s Z rotation by either deactivating its up-vector constraint or modifying its Roll parameter. For more information, see Up-vector Constraints on page 312.
Keying Transformations Quickly

The best method for quickly keying all axes for each type of transformation is to use the keyboard shortcuts found on the number pad (at the right of your keyboard), as mapped out here:

- **Scaling X, Y, Z** Press 7
- **Rotation X, Y, Z** Press 4
- **Translation X, Y, Z** Press 1

Removing Transformations

To remove transformations

1. Select one or more objects with transformations.

2. From the Animation panel, choose Animation > Remove Animation > from Transforms to remove all scaling, rotation, and translation parameters.

You can also choose to remove each of these types of transformations (X, Y, and Z axes) individually with their appropriate commands.
Chapter 6 • Animating Transformations

Animating Transformations in Hierarchies

Transformations are propagated down through hierarchies. Each object’s local position is stored relative to its parent. It’s as if the parent’s center is the origin of the child’s world.

Objects in hierarchies behave differently when they transformed, depending on whether the objects are node-selected or branch-selected. By default:

- When you branch-select a parent object and animate its transformation, the animation is propagated to its children.
- When you node-select a parent and animate its transformation, its children are not transformed unless their respective local transformations are animated. For example, suppose the child’s local translation is animated but its rotation isn’t: if you translate the parent, the child follows; however if you rotate the parent, the child stays in place.

This is because animation on the local transformations is stored relative to the parent’s center. You can make unanimated children follow the parent with the Constrain > Child Transform Compensation command (or ChildComp button) on the Constrain panel (see Transformations and Hierarchies on page 79 in the Transformations guide).

- When you animate a child object, its animation is always done relative to its parent (local animation).
- When you animate anything in global, it’s always done in relation to the world origin: it does not matter if your objects are in a hierarchy or not. Nothing is inherited if you have global transformation keys because they override any parent-to-child inheritance.

Chains are an exception because the end location of one element always determines the start location of the next one in the chain.

Changing the Animation Inheritance

By default, children follow the parent when branch-selected, but you can change all that.

The Active option on the Options page of the Local Transform property editor turns this parent constraint on or off. When this option is off, the object does not follow its parent when the parent is transformed (does not inherit its transformations), not even when the parent is branch-selected nor when Child Transform Compensation is off.
Keeping the Active option selected, you can also toggle the Constrain > Position, Orientation, and Scaling options to determine each one's inheritance individually.

One reason to disconnect the inheritance is if you always want to animate an object globally. If you deselect the Active option, the global transformation parameters (instead of local) are marked automatically when you activate a transformation manipulation mode because the object does not inherit the parent's transformations. Then it's easy to key the global parameters immediately.

For example, you cannot key the global position or orientation of bones, but if you wish to animate a bone globally, you can prevent the bone from inheriting its parent's position or rotation by deselecting the appropriate Constrain options (Position or Orientation).

Only the specific type of transformation that's being affected is considered, which is the same as it works in SOFTIMAGE|3D. For example, suppose the child's local translation is animated but its rotation isn't. If you translate the parent, the child follows; however if you rotate the parent, the child stays in place.
Creating Transform Groups (Parenting with Null Objects)

When you’re working with hierarchies, especially characters, you often create a null to be a parent of the current selection or the whole hierarchy. While you can use any object as a parent, nulls are often used because they have no geometry and are therefore not modeled or rendered.

To make this parenting process easy, you can create a transform group, which is essentially a null that becomes parent of the selected objects. This null is not visible in a 3D view, by default.

To create a transform group

1. Select the objects or branch-select the hierarchy or model that you want to be in the transform group.

2. Choose Edit > Create Transform Group or press Shift+/. If you are using the QWERTY keyboard mapping, press Ctrl+g.

A null called transfogroup is created at the location of the selected objects’ parent, and becomes the new parent of the selected objects. If the selected objects have no parents, or if they have different parents, the transform group null is placed at the scene origin.

The transform group is contained within the current hierarchy or model of the selected objects. If the selected objects aren’t within a model, the transform group is created under the scene root node.

If the selected objects aren’t in the same hierarchy, the transform group is created within their common model; and if the selected objects aren’t in the same hierarchy or model, the transform group is created under the scene root.

If you want to scale, rotate, or translate the transform group null while keeping it hidden, select it in the explorer, choose Transform > Transform Hidden Objects, and transform as usual.

To display the transform group null

- Click the eye icon (visibility menu) in a 3D view and select Transform Groups, or select it in the Camera Visibility property editor (press Shift+s).

You can change the size, shape, and add a shadow for a null in the same way as you can for any skeleton chain element. See Creating Shadows and Changing the Chain Element Display on page 48 for more information.
Creating Transform Groups (Parenting with Null Objects)

To remove a transform group

- Select the transform group null and choose Edit > Remove Transform Group or press Ctrl+Shift+/

This deletes the transform group null, then reparents its children “up a level” to the model in which the transform group was contained, much like the Cut command does for removing hierarchy relationships. If there is no parent, the children are put under the scene root node.

Don’t press the Delete key to delete the transform group! If you branch-select the transform group null and press Delete, all objects in its hierarchy are also deleted as they would be in a typical hierarchy.
Chapter 6 • Animating Transformations

**Animating the Transformation of Object Centers**

Although you can transform an object’s center in Center mode (see *Center Manipulation* on page 64 in the Transformations guide), you cannot directly animate its transformation. You would want to animate an object’s center to, for example, make an object rotate from a given pivot and then have the pivot change to somewhere else relative to the geometry.

You can do this in XSI by animating the object pivot as described in *Working with Object Pivots* on page 66 of the Transformations guide.

Another way is to constrain the object’s center to a null using a Pose constraint (or make it a child of the null), then animate the null’s transformations. The Pose constraint constrains the scaling, rotation, and translation values all together. For more information, see *Pose Constraints* on page 300.

You can also constrain the object’s center to two different nulls, animate the nulls, then blend the nulls’ constraint weights to control how much the object is influenced by each null. As well, you can animate this weight blending. For more information, see *Blending Constraints* on page 321.
Animating Rotations

Rotations have some special considerations more than the other transformations. Some of the issues that you need to understand when animating rotations are the transformation node in which you can manipulate and key rotations (see Manipulation Modes versus Transformation Values on page 143), Euler versus quaternion rotations, gimbal lock, and continuous rotations.

Euler versus Quaternion Rotation

When you animate rotations in XSI, you normally use three separate function curves that are connected to the X, Y, and Z rotation parameters. These three rotation parameters are called Euler angles. Euler angles are simple to understand and control because each rotation axis is evaluated in a set order. This process is known as Euler interpolation.

Euler interpolation works well when the axis of interpolation coincides with one of the XYZ rotation axes, but is not as good at interpolating arbitrary orientations. For example, it may be difficult to get smooth interpolation when animating the orientation of either a ball joint, such as a shoulder, or of a tumbling object whose animation is not around one of the XYZ axes.

Euler angles can also suffer from gimbal lock, which is the phenomenon of two rotational axes aligning with each other so that they both point in the same direction. This happens because the XYZ axes are not independent, so changing one axis can affect the preceding ones. For example, if you apply a 90-degree rotation to the Y axis, it causes the X axis to coincide with the Z axis so that rotating in X and Z has the same effect.

Gimbal lock can be compensated for by changing the order in which the rotation axes are evaluated (by default, it’s XYZ), which changes where the gimbal lock occurs (see Changing the Bones’ Order of Rotation on page 75 in the Character Animation guide).

About Quaternion Interpolation

Quaternion interpolation provides a way to overcome these limitations and provide smooth interpolation with any sequence of rotations. With this type of interpolation, the XYZ angles are treated as a unit to determine an object’s orientation so they are not restricted to a particular order of rotation axes. Quaternions interpolate the shortest path between two rotations.
Chapter 6 • Animating Transformations

Quaternion Function Curves

You can create quaternion fcubes in XSI by either setting quaternion keys or converting Euler to quaternion function curves. These curves are true quaternion curves, but are drawn as three separate fcubes (as Euler fcubes are represented) in the fcurve editor to make them easier to edit.

Because the three fcubes are an approximation of the underlying quaternion fcube, the displayed values may have discontinuities where the XYZ angles change abruptly. This is only a display artifact and does not affect the interpolation. You can usually minimize these discontinuities by making the fcurve rotations continuous (see Making Rotations Continuous on page 156).

When you set, delete, or move a key on a quaternion fcurve, the change appears on all three fcubes because the entire rotation is being interpolated by the quaternion fcurve.

You can edit quaternion fcubes in both the fcurve editor and the dopesheet as you would normal fcubes. However, quaternion fcubes do not have tangents for editing fcurve slopes as typical fcubes do, but instead use special parameters to edit the fcurve's interpolation (see Editing Quaternion Function Curves on page 226).

It's a simple process to convert from Euler to quaternion rotations and back again. However, as with most conversions, information is changed in the process. This is because quaternion fcubes store one key for the entire XYZ rotation value.

When you convert Euler fcubes to quaternion fcubes, a quaternion key is added for each key in the XYZ fcubes. For example, if there is a key at frame 5 for the Euler rotation X parameter but not Y and Z, a quaternion key is generated for each rotation parameter by sampling the XYZ values at frame 5.
Animating Rotations

Also, any slope (tangent) information on the Euler fcurves is lost when converting to quaternions.

If you want to preserve the original rotation fcurves, you could store them in an action source. You can then apply the source back to the object if you want to return the original fcurves. See Actions on page 89 in the Nonlinear Animation guide for information.

**To convert Euler to quaternion rotation**

1. Select an object with Euler rotation fcurves and choose Animation > Convert Euler Rotation to Quaternion from the Animation panel.

2. Edit the timing and interpolation of the quaternion curves in the fcurve editor using the Quaternion Key controls—see Editing Quaternion Function Curves on page 226.

**To convert quaternion to Euler rotation**

1. Select an object with quaternion fcurves and choose Animation > Convert Quaternion Rotation to Euler from the Animation panel.

2. Edit the Euler function curves as you would normally.

**Keying Quaternion Rotations**

Setting keys to create quaternion rotation fcurves is the same process as for setting Euler rotation fcurves except that you can key quaternion rotations only on an object that already has rotation fcurves (either Euler or quaternion). When you set and key the object’s rotation as quaternion, you get a smooth interpolation.

**To create quaternion rotation fcurves**

1. Select an object with either Euler or quaternion rotation fcurves.

2. Choose Animation > Convert Euler Rotation to Quaternion from the Animation panel.
   
   If you select an object that already has Euler rotation fcurves, the curves are simply converted to quaternion.

3. Key rotations as usual (see Keying Transformations on page 146) and the resulting fcurves will be quaternion.

   Setting keys for one rotation axis creates corresponding keys on the other two rotation axes.
4. Edit the timing and interpolation of the quaternion curves in the fcurve editor using the Quaternion Key controls—see Editing Quaternion Function Curves on page 226.

**Making Rotations Continuous**

When you animate rotations 360 degrees or more, there may be problems with the rotations flipping or “spiking” when the values jump between equivalent but discontinuous numbers. For example, if you’re rotating a wheel 360° in Y, and it goes 180° in the correct direction but then rotates backwards to its original starting value.

By making a rotation fcurve continuous, the flipping is eliminated. If two keys on the fcurve have a gap of more than 180°, the second key is offset by 180° so that the object’s rotation continues in the same manner.

You may want to make rotations continuous if you’re planning to stretch, mix, or manipulate the animation as an action in the animation mixer.

**To make rotations continuous**

- In the fcurve editor, select an fcurve and choose Curves > Make Rotations Continuous from the fcurve editor command bar.

**Making Rotation Keys Continuous**

If you want to process more than one rotation fcurve at a time, you can choose the Make Rotation Keys Continuous command from the Animation panel. This command processes all rotation curves on an object together as a complete orientation instead of individual X, Y, and Z rotation function curves.

This command also processes the local and global rotation keyframes on the selected objects to minimize the change between successive Euler angle rotation keyframes, making the values continuous between keys.

**To make rotation keys continuous for fcurves**

1. Select an object that has rotation fcurves (either Euler or quaternion).

2. Choose Animation > Make Rotation Keys Continuous from the Animation panel at the bottom of the XSI window.
Visual Cues for Animated Transformations

If any of an object's transformations (scaling, rotation, and translation) are animated, there are visual indicators (colored triangles) in the viewport that let you know how they're animated: by fcurves (keys), expressions, scripted operators, or an action clip in the animation mixer.

Not only can you see how objects are animated, you can immediately access the animation information itself. For example, if a selected object has an expression on it, you can click the animation cue triangle, select the parameter from the list that appears, and the expression editor opens up with that expression.

**To view the animation cues**

1. Select one or more objects whose transformations are animated. If only one object is selected, the triangle size is larger for easier picking.
2. Click the eye icon in the viewport's menu bar and choose Animation Cues.

A triangle is drawn on the appropriate object to show that it is being driven by a particular kind of animation. The different colors and locations of the triangles indicate the type of animation by which the object's parameters are being driven:

- **Function curves**—a downward-pointing triangle is shown above the selected object's center. The triangle is:
  - Red if there is a key at the current frame.
  - Green if not at a keyframe.
  - Yellow if any parameters have been manually altered to be different from the fcurve value at the current frame.

- **Expressions**—a right-pointing triangle is shown to the left of the object's center.

- **Scripted operators**—a left-pointing triangle is shown to the right of the object’s center.

- **Action clips**—an upward-pointing triangle is shown below the selected object's center. The triangle is:
  - Gray if no actions are currently driving the parameters.
  - Pink if only one action is driving at the current time.
  - Red if more than one action is driving at the current time.
Chapter 6 • Animating Transformations

To access animation information

- Click the triangle and select a parameter from the list.
  - For function curves, expressions, and scripted operators, the animation editor opens with the appropriate type of editor active and with the relevant parameter selected.
  - For action clips, selecting a parameter and one of its action clips selects that action clip in an open animation mixer.

If there are multiple clips driving a parameter, the number of clips currently driving the parameter is indicated, as well as the total number of clips driving the parameter. For example, name of parameter 2/5 means that there is a total of 5 clips driving this parameter, but only 2 are driving at the current time.

You can choose the clip by clicking the arrow beside the parameter. The clips that are active at the current frame have an asterisk (*) after their name.

If you rectangle-select all the cues, a list of all parameters associated with those animation cues appears.
  - An F appears in front of the parameter names for fcurves.
  - An M (for mixer) appears in front of the parameter names for action clips.
Chapter 7  Editing Function Curves

When you set keyframes to animate a parameter, a function curve is created. A function curve, or fcurve, is a representation of the animation values as a graph in the fcurve editor.

- Time is shown plotted along the X axis (horizontal).
- The parameter’s value is plotted along the Y axis (vertical).

The shape of the fcurve shows how the parameter’s value changes over time.

![Function curve for bouncing ball in the fcurve editor below shows its animation on its Y axis.](image)

On the fcurve, keyframes are represented by key points (also referred to as simply keys) and the interpolation between them is represented by segments of the curve linking the key points.
The handles of each key let you define the fcurve’s slope in the same way that control points define Bézier curves, while the slope of the curve determines the rate of change in the animation. For example, if the slope inclines toward a key, the animation is accelerating; a decline means a deceleration.

**Working in the Animation Editor**

The animation editor is where you control the animation of the currently selected element using a number of different editing tools.

- **Command bar**
  Contains commands and tools for saving, editing, and viewing animated properties (such as fcurves).

- **Animation tree**
  Displays the animatable properties of selected elements.

- **View area**
  Displays and allows manipulation of property animation using the graph (for fcurves, as shown here), the dopesheet, the expression editor, or the scripted operator editor.

**Opening the Animation Editor**

There are several ways to display the animation editor:

- Choose the Animation Editor view from a viewport title bar. Depending in which viewport you made your selection, the animation editor is anchored across the two upper or lower viewports.

  or

- From the main menu, choose View > Animation > Animation Editor. The animation editor is displayed in its own floating window, which you can move and resize like any other window.

  or
Working in the Animation Editor

- Select an object and press 0 (zero) to also open the animation editor in a floating window. The selected object’s parameters are displayed automatically.

or

- In the explorer, a property editor, or the weight panel in the animation mixer, right-click an animation icon (green box) beside a parameter and choose Animation Editor, as shown on the left. Again, the animation editor is displayed in its own floating window.

or

- In the explorer, click the animation icon of a parameter animated by a function curve, expression, linked parameter, or scripted operator. The corresponding editor (fcurve, expression, or scripted operator) appears in the animation editor automatically.

The animation editor responds to the currently selected elements. While it is open, it displays the animation of the elements you select.

To open the animation with specific elements already selected, see Opening the Fcurve Editor with Specific Fcurves Selected on page 168.

The view area (right panel) can display the fcurve graph (default), the dopesheet, the expression editor, or the scripted operator editor. If the animation editor is in a floating window, the name of the current editor is displayed in the window’s title bar.

Selecting the Animation Editing Tool

To change editors in the view area

- Choose the appropriate command from the Editor menu on the animation editor’s command bar:

  - Choose Fcurve Editor to display the fcurve view. You can also select a parameter that is animated with a function curve to display this view.

  - Choose Dopesheet to display the dopesheet, which displays animation as segments on tracks. This gives you an overall view of your animation, making it easy to do tasks like scaling or offsetting animation.

  For more information, see Chapter 8: Editing Animation in the Dopesheet on page 241.

  - Choose Expression Editor to display the expression editor. This editor is also displayed automatically when you select a parameter that has an expression applied to it.

  For more information, see Chapter 12: Animating with Expressions on page 345.
Chapter 7 • Editing Function Curves

- Choose **Scripted Operator Editor** to display the scripted operators editor. This editor is also displayed automatically when you select a parameter that has a scripted operator applied to it.

  For more information, see **Scripted Operators** on page 151 in the Customization guide.

- Middle-click the **Editor** menu name to return to the previous editing tool that was displayed.

### Using the Animation Tree

The animation tree in the animation editor is actually an explorer (see *The Explorer* on page 83 of the Interface and Tools guide for general information). It displays the animatable properties of the currently selected elements (these have the animation icon beside their name).

The View, Lock, Update, and Synchronize buttons in the command bar apply only to the animation tree, regardless of the animation editing tool shown in the view area.

For information on using the View menu and the Synchronize button, see *Controlling Which Fcurves Are Displayed* on page 174.

The parameters in the tree are similar to what’s shown in a “regular” explorer, but in this tree, the script name of the parameter is used. For example, the tree on the left shows the object name (*disc*) and its animated parameters beneath it (*kine*, then *local*, and *rotx* selected).

You can use the tree to choose which properties (parameters) to display in any of the animation editing tools shown in the view area, such as the fcurve graph.

**To select parameters in the tree**

- Click the name of the parameter so that it is highlighted in purple.
- Select a branch of parameters by clicking on their parent node. For example, click the *local* node to select all position, rotation and scaling parameters (as shown on the left).

  For more information on specific animation parameters in the tree, see *Finding Animation Elements* on page 30.

- You can also drag-and-drop elements from the explorer or any other view to this tree—make sure to click the **Lock** button before doing this so that the animation tree doesn’t update to the new element you have selected.

**To expand and collapse an animation tree**

- To **expand** an item in an animation tree, click its plus sign (+). The elements, properties, and subproperties below it are listed.
- To **collapse** an item in an animation tree, click its minus sign (−). The items below it are no longer listed.

- Choose **View > Expand All** to expand all nodes of the tree.
To lock and refresh the tree

- Click the Lock icon on the command bar to “lock” the selected tree view and prevent it from updating (if you select a different object).

Click the button again to deactivate it (and update the tree if the selection has changed).

- To update to a new object while keeping the animation tree locked, select the object and click the Update icon.

This refreshes the animation tree contents to the currently selected object.

To reset the tree’s layout

- Choose View > Reset or press r. This resets the layout of the animation tree to what it was when you first opened it.

To hide the tree

- Click the little arrow on the bar that divides the tree from the view area (fcurve graph, expression panel, etc.). Click the arrow to redisplay the tree.

To display parameter names in color

- Choose View > Use Wire Colors. Transformation parameters in X are red, Y are green, and Z are blue. Alpha parameters are light blue. When you select an fcurve, the unselected keys on that curve match the parameter color.

- You can also set custom colors for the parameters as you like by right-clicking a parameter in the animation tree and choosing Set Parameter Color. This opens a color editor in which you can set a color that overrides the default colors for the parameters.

These colors appear for the parameter names in the animation tree, on their fcves in the fcurve graph, and their keys in the dopesheet. Make sure that the View > Use Wire Colors command is active to see the colors.

Setting Preferences for the Animation Tree

The animation editor preferences let you set up options for the animation tree in the animation editor. These preferences apply to the animation tree regardless of the current editor displayed in the view area: fcurve editor, dopesheet, expression editor, or scripted operator editor.
Chapter 7 • Editing Function Curves

There are separate preference property editors for the fcurve editor (see Setting Fcurve Editor Preferences on page 166) and the dopesheet (see Setting Preferences for the Dopesheet on page 243).

To set the animation editor/tree preferences

1. Choose File > Preferences from the XSI main menu or choose View > Preferences for the animation tree.

2. In the explorer that appears, click the Editors > Animation Editor icon or node.

3. Set the following preferences for the animation tree:
   - **Track Selection**: Automatically ensures that the first selected node is visible in the animation tree.
   - **Parameter Sorting Method**: The sorting method used by the animation tree parameters: None (creation), Alphabetical, or Layout.
   - **Max. Number of Curves to Display**: the maximum number of curves to load in the animation editor when opening it or changing the selection. This can help prevent a delay when loading a large number of fcurves, such as with a branch-selected object.

   A value of zero (0) displays all curves. Although the slider goes only to 100, you can enter a value up to 30,000.

Using the Animation Editor’s Timeline

By default, the red playback cursor is displayed in the animation editor’s timeline and over the view area. This lets you easily scrub through the animation and go to the exact frame you want.

As well, you can see and move loop markers in the animation editor’s timeline when you click the Loop button in the playback panel.

To hide the playback cursor

- Deselect View > Time Cursors Visible.
To set a preference for displaying the playback cursor

1. Open the Fcurve Editor Preferences editor (see Setting Fcurve Editor Preferences on page 166).

2. On the View page, toggle the Time Cursors Visible option. This setting controls all time-related cursors in the animation editor (current frame, key time, etc.).

   Toggling the command in either the View menu or the Fcurve Editor Preferences editor changes its state in the other.

By default, you can drag the cursor in the timeline area but not in the fcurve graph area. However, you can change the behavior so that you can also select and drag the playback cursor in the graph area.

Note that the playback cursor is always available in the timeline area, as long as it’s visible.

To make the playback cursor available in the graph

- Choose View > Time Cursors Selectable.

To set a preference for making the playback cursor available

1. Open the Fcurve Editor Preferences editor.

2. On the View page, toggle the Time Cursors Selectable option.

   Toggling the command in either the View menu or the Fcurve Editor Preferences editor changes its state in the other.

To copy key values to another frame

- Right-click+drag the cursor in the animation editor’s timeline to copy key values from one frame to another, just like you do in the main XSI timeline (see Keying the Current Values at a Different Frame on page 135).

   The playback cursor stays green until you set a key.
Chapter 7 • Editing Function Curves

Working in the Fcurve Editor

There are many ways of setting up the fcurve editor so that you can work in it as you like. You can set a number of preferences, hide or access certain commands, click icons instead of choosing commands, or have the fcurve editor open up with the elements you want selected. As well, there are a few ways in which you can easily find elements animated with fcurves, or get fcurve information.

Setting Fcurve Editor Preferences

The fcurve editor preferences let you set up options for different tasks you do in the fcurve editor, such as setting keys, setting the slopes of the fcurves, or displaying audio waveforms and markers.

There are also preference property editors for the dopesheet (see Setting Preferences for the Dopesheet on page 243) and the animation tree in the animation editor (see Setting Preferences for the Animation Tree on page 163).

To set the fcurve editor preferences

1. Do either of the following:
   - Click the Preferences icon in the fcurve editor command bar.
   - Choose File > Preferences from the fcurve editor's command bar.
   - Choose File > Preferences from the XSI main menu. In the explorer that appears, select Editors > Fcurve Editor or click its icon.

2. In the Fcurve Editor Preferences property editor, you can set the default preferences.

   Each option is discussed in its appropriate section in this chapter (for options on the Audio page, see Viewing Audio Waveforms on page 210 in the Nonlinear Animation guide).

   You can also check the online help (click the ? icon) in this property editor for information on each option.

Commands and Tools for Editing Fcurves

The fcurve editor command bar contains the menu commands and tools you need for editing fcurve animation.

Context Menus

You can access many of the commands in the command bar in context menus, depending on what's currently selected.

• To open context menus for fcurves, select an fcurve and either right-click or press Alt+right-click.
• To open context menus for keys, regions, and fcurve slopes, select the appropriate elements and press Alt+right-click.

**Hiding Parts of the Command Bar**

You can hide certain parts of the command bar that are optional. For example, the Navigation panel containing the zoom, pan, and frame icons is hidden by default.

**To hide or display areas of the command bar**

• Right-click anywhere in an empty area at the top of the window and toggle the appropriate panel on or off (a check mark means that it’s displayed).

The following image shows the name of each panel on the command bar.

**Icons**

A number of commonly used commands in the menus on the command bar are also represented as icons for quick access.
Undoing and Redoing Fcurve Modifications

Like any other operation in XSI, you can use the standard shortcut keys to undo or redo fcurve modifications:

- To undo any modifications to an fcurve, press Ctrl+z.
- To redo the modification you just undid, press Ctrl+y.

You can also use the Snapshot and Swap icons in the fcurve editor command bar to keep track of the changes you’ve made to an fcurve as you’re working. Snapshot keeps a copy of the “original” curve before you made modifications, allowing you to revert to this curve. For more information on these, see Editing Safely with Snapshots of Function Curves on page 184.

Opening the Fcurve Editor with Specific Fcurves Selected

When you select an object and open the animation editor, you can view all fcurves created for that object. However, you usually want to hone in on specific curves. Two common elements you want to have easily available are marked parameters and transformation parameters.

To open with marked parameters selected
- Mark some parameters and then open the animation editor. The marked parameters are automatically added to the selection in the editor.

For example, if you have the kine.local.pos parameters marked and you open the animation editor, it automatically selects the marked parameters (in the tree and graph).

To open with transformation parameters selected
- Do one of the following:
  - Press Ctrl+0 to open with all local transformation parameters selected.
  - Press the following key combinations on the key pad to open the animation editor with the individual local transformation parameters selected (make sure that NumLock is on).

<table>
<thead>
<tr>
<th>X</th>
<th>Y</th>
<th>Z</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ctrl+7</td>
<td>Ctrl+8</td>
<td>Ctrl+9</td>
</tr>
<tr>
<td>Ctrl+4</td>
<td>Ctrl+5</td>
<td>Ctrl+6</td>
</tr>
<tr>
<td>Ctrl+1</td>
<td>Ctrl+2</td>
<td>Ctrl+3</td>
</tr>
</tbody>
</table>
Selecting Objects with Fcurves

When you have a crowded scene, it's sometimes difficult to select only the objects you want. To make it easier to select only objects that are animated with function curves, you can use a special fcurve filter from the Filters menu on the Select panel.

To use the selection filters
1. In the Select panel of the main command panel, click the little arrow button to open the Filters menu.
2. Select the Obj w Fcrv Anim filter.

Getting Fcurve Information

You can use a special query in the spreadsheet to show the fcurves for an object. This query makes it easy to get an overview of all the information about each fcurve for the selected objects.

To select the spreadsheet query
1. Select one or more animated objects.
2. Open a spreadsheet (press Alt+3) and choose Query > Fcurves.

<table>
<thead>
<tr>
<th>Name</th>
<th>Fcurves</th>
<th>Number of keys</th>
<th>Value when no key</th>
<th>Curve type</th>
<th>Extrapolation</th>
<th>Sequence Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>soccerBall</td>
<td>local_posx</td>
<td>5</td>
<td>-0.1273041690135417</td>
<td>Standard</td>
<td>Constant</td>
<td>Spline</td>
</tr>
<tr>
<td></td>
<td>local_posy</td>
<td>5</td>
<td>0.15710077397776455E-16</td>
<td>Standard</td>
<td>Constant</td>
<td>Spline</td>
</tr>
<tr>
<td></td>
<td>local_posz</td>
<td>5</td>
<td>-5.52332094219454</td>
<td>Standard</td>
<td>Constant</td>
<td>Spline</td>
</tr>
</tbody>
</table>

For more information, see The Spreadsheet on page 106 in the Interface and Tools guide.

Working with High-Density (Raw) Fcurves

High-density fcurves are curves that have many keys, usually one per frame. These curves are usually the result of importing motion-capture animation or from plotting an animation.
Because editing high-density curves is not always an easy task, there are a number of tools in the fcurve editor that can help you work with them:

- Use the HLE (high-level editing) tool for editing a high-density curve’s shape via a “sculpting” curve—see *Shaping High-Density Fcurves* on page 229 for more information.

- Resample, smooth, or fit curves to reduce the number of points on a curve while keeping its shape—see *Processing Function Curves* on page 233 for more information.

- Convert a raw fcurve to a standard fcurve—see *Converting Between Raw and Standard Fcurves* on page 237.

- Use the animation tree on the left of the fcurve editor to select a specific parameter and its curve instead of trying to select the curve directly in the graph—see *Using the Animation Tree* on page 162 for more information.
Setting Up the Fcurve Graph

When the animation editor is open with Editor > Fcurve View active, it displays the fcurves of any elements you select.

The graph in the view area is where fcurves are displayed and edited. You can set up the graph to display and set up the grid and axis element as you like.

Setting the Time Format (X Axis)

The graph's horizontal axis (X) displays the time scale in frames or milliseconds, while the vertical axis (Y) displays the values of the animated parameter.

Setting the main XSI timeline's format also sets the format for the timeline in the animation editor, as well as the dopesheet and the animation mixer. See Setting the Timeline Display Format on page 78 for more information on how to set the animation editor's timeline.

Setting the Graph's Grid and Rulers

To show or hide the grid

- Choose View > Grid from the fcurve editor command bar.
  
  or
  
  - Press g.

To selectively show or hide the X-axis or Y-axis grid lines

1. Open the Fcurve Editor Preferences editor (see Setting Fcurve Editor Preferences on page 166).
2. On the Editor page, select Display Grid > Show for the X and/or Y axes.

To set the size of units for the grid

- Specify a value in the Display Grid text boxes for the X and/or Y axes.
To show or hide the X-axis and Y-axis rulers

- You can display or hide both the X and Y axes (the numerical values) of the graph by choosing View > Rulers from the command bar.

To selectively show or hide the X-axis or Y-axis rulers

- On the Editor page in the Fcurve Editor Preferences property editor, select Rulers > Show for the X and/or Y axes.

To set the units for the ruler

- Specify a value in the Ruler text boxes for the X and/or Y axes.
Viewing Function Curves

When the animation editor is open with Editor > Fcurve View active (the default), it displays the fcurves of any elements you select.

The graph in the view area is where you manipulate the fcurve of any animatable parameter. You can edit the keyframed values by adjusting the keys on a selected curve.

The graph’s horizontal axis (X) displays the time scale in frames or milliseconds. The vertical axis (Y) displays the values of the animated parameter.

Curves for X, Y, and Z parameters are red, green, and blue, respectively. For example, an fcurve for scaling in Z is blue, whereas scaling in X is red.

To set custom colors for fcurves, see Using the Animation Tree on page 162.

**Key coordinates** indicate the exact frame number (on the X axis) and value (here on the Y axis) of the most recently selected key.

**Keyed values** on fcurves are indicated by **keys**. Selected keys are red, and unselected keys are blue.

**Values** for the parameter are shown on the Y axis.

**Time** is shown on the X axis.

**Curve and axis labels** indicate the type of curve and the affected axis. For example, this fcurve represents the object's rotation on the Z axis. The axis label is in the graph's lower-left corner.

The **slope handles** at each key indicate the rate at which an fcurve's value changes. You can change the curve's slope by manipulating these handles.

By default, fcurves use **spline interpolation** to calculate intermediate values.
**Controlling Which Fcurves Are Displayed**

Use the commands in the two View menus to determine what you want to display in the animation tree and the graph, respectively. You can control how fcurves are displayed, such as:

- Which parameters are shown in the animation tree.
- Which types of fcurves to display in the graph.
- What sort of information each fcurve displays.

Only nodes that lead to a visible animated parameter are shown in the animation tree, regardless of its View menu filter (colors only, kinematics, etc.). Children objects are shown only if the branch is selected. Root objects are always shown because they define the scope of the view.

**Choosing Which Parameters Are Displayed in the Animation Tree**

To determine what is displayed in the animation tree, you use the commands in the View menu on the furthest left. Many of these commands are the same as found in the explorer, as described in *Sorting and Reordering Objects in the Explorer* on page 90 of the Interface and Tools guide. The following commands, however, are specific to the animation tree and determine the effect of the rest of the filters in the menu:

- **Animated Parameters Only** is especially useful for keeping the animation tree uncluttered. It displays only the element's animated parameters and their associated fcurves. This is on by default when you open the animation editor.
- **Activation Params Only** displays any activation parameters (the Active or Mute parameters) and their associated fcurves. These are parameters whose script name matches `active` or `mute`. They appear under nodes that have a Mute parameter (such as deformations), or constraints and expressions which have an Active parameter.
- **Selected Parameters Only** displays only selected parameters for the element and their associated fcurves. You can narrow down the selection even more by selecting parameters from this list and clicking the Synchronize icon in the toolbar to update with this information.

For example, let’s say you had 10 parameters initially displayed when you chose Selected Parameters Only. From these 10, you selected only five parameters and clicked the Synchronize icon to update with only this information. To return to the initial selection, deselect everything and click the Synchronize icon.

The Synchronize icon is active only when Selected Parameters Only is active.
• **Marked Parameters Only** displays only the marked parameters for the element and their associated fcurves. This option is mutually exclusive with **Animated Parameters Only**. Changing the marking automatically refreshes the tree.

• **Driving Actions at Curr Time** displays parameters that are driven by an action clip in the animation mixer at the current frame (the playback cursor is at a frame that is covered by an action clip). Only the fcurves for the selected object’s current clip are displayed in the fcurve graph.

• The **SI|3D Style** option sets different options for you so that you can work in the animation editor in a way similar to SOFTIMAGE|3D. These options are activated when you select **SI|3D Style**:
  - View > Flatten All
  - View > Use Script Names
  - View > Animated Parameters Only

• **Locked Parameters** displays locked parameters for the element and their associated fcurves.

• **Tagged Parameters** displays tagged parameters for the element and their associated fcurves. You can select to show All tagged parameters, or only parameters tagged with certain tag names (Tag 1 through Tag 10).

• The rest of the commands in the View menu below these ones let you display the fcurves or tracks only for the elements as their names indicate, such as Position, Rotation, or Scaling Only. What is shown with these filters depends on which parameter you have selected. For example, if **Animated Parameters Only** is selected and you choose the Rendering Properties filter, but there are no animated rendering parameters, no parameters are shown in the animation tree.

**Choosing Which Fcurves Are Displayed in the Graph**

Once you have the parameters you want shown in the animation tree, you can control the display of fcurves in the graph with the commands in the View menu for the fcurve graph:

• **View > All Parameters** displays the fcurves of all animatable parameters for the selected object. You can also press 6.

• **View > Animated Parameters** displays the fcurves of all animated parameters for the selected object. You can also click its icon in the toolbar or press 7.

• **View > Marked Parameters** displays only the fcurves of marked parameters for the selected object. You can also click its icon in the toolbar or press 8.
Chapter 7 • Editing Function Curves

View > Marked Parameters does not apply to marking sets, only to parameters that are actually marked (see Marking Parameters on page 36 to know which parameters are marked). Marking sets are actually custom parameter sets that contain proxy parameters.

To see marking sets, choose View > Custom Properties for the animation tree and make sure that View > Animated Parameters Only is off. You can also choose the flattest display mode, View > Flatten All. Now in the tree, you will see the marking set with its parameters beneath it.

• View > Selected Parameters displays only the fcurves of selected parameters for the selected object. You can also click its icon in the toolbar or press 9.

To display fcurves for unanimated parameters

1. Make sure that View > Animated Parameters Only is off in the animation tree.
2. Choose View > Selected Parameters for the fcurve graph.
3. In the animation tree, browse for and select the unanimated parameters for which you want an fcurve.

Selecting a folder (such as kine.local) generates fcurves for all unanimated parameters within it.
4. A flat fcurve appears for each unanimated parameter you select. You can then add keys to this fcurve and edit it as you like.

Locking on to an Element

When you lock an element, its fcurves are always visible. You can then select other elements and still have access to the locked element’s fcurves in the fcurve editor.

Click the Lock icon on the command bar to force the selected view to stay open (“locking” it) and prevent it from updating if you select a different object. Click the button again to deactivate it (and update the animation tree if the selection has changed).

When the animation tree is locked, you can drag-and-drop new objects into it from any geometry view or an explorer.

To update to a new object while keeping the animation tree locked, select the object and then click the Update icon. This refreshes the animation tree contents to the currently selected object.
Hiding and Displaying Fcurves

You can hide one or more curves in the graph. Hiding curves does not remove them from the view, so any snapshot information is retained and available once you display the curve again.

To hide fcurves

- Do one of the following:
  - Select one or more curves and choose View > Hide Curve to hide them or press h. This also deselects the curves.

  or

  - Choose View > Clear All Curves to hide all fcurves associated with the selected object without collapsing the animation tree.

To display all hidden fcurves

- Do one of the following:
  - Choose View > Unhide All Curves or press Shift+h.

  or

  - Select the parameter name from the animation tree.

Hiding and Displaying Fcurve Information

You can choose to view extra curve, key, and slope information while you’re working in the graph. However, if you have many fcurves displayed in your graph, you may want to hide a certain amount of curve information to keep the view uncluttered.

To hide or display curve information

From the command bar, choose any or all of the following:

- View > Keys on Unselected Curves to show or hide keys on all unselected curves in the graph. If you choose this command, you can select keys without having to first select the fcurve.

  If you don’t choose this command, you must select an fcurve before being able to select keys on it. This prevents you from accidentally manipulating keys, which can be a problem if there are many curves and keys.

  You can set the default for this option’s state (on or off) on the View page in the Fcurve Editor Preferences property editor—see Setting Fcurve Editor Preferences on page 166.
Chapter 7 • Editing Function Curves

- **View > Slopes on Unselected Keys** to show or hide the slope handles of all unselected keys on a selected curve.

  You can set the default for this option’s state (on or off) on the View page in the Fcurve Editor Preferences property editor.

- **View > Keys Coordinates** to show or hide the current time and value of a selected key.

  You can set the default for this option’s state (on or off) on the View page in the Fcurve Editor Preferences property editor.

- **View > Curve and Axis Labels** to show or hide labels on all selected curves in the graph. Because fcurves for expressions and linked parameters do not necessarily use the same horizontal time units as other curves, selecting Curve and Axis Labels helps you to know the meaning of their values.

### Viewing Fcurves of Different Values (Normalizing)

When you’re working with many fcurves, it’s not unusual that some curves will have varying values: one curve is too big to fit in the graph, while another is too small to see clearly. By “normalizing” the graph’s Y axis, you can easily view multiple curves that have a broad range of values. Normalizing scales all selected curves so that they fit in the graph, regardless of their unit scale.

**To normalize the Y axis**

1. Select the curves you want to fit in the graph view.
2. Choose View > Normalized in Y (Value) from the fcurve editor command bar to activate it. To return to the default linear scale, choose it again.

For example, let’s say you have one curve that spans values between 0 and 5, and another curve whose values range between 0 and 400. Viewing both of them on a standard scale would make the first curve very difficult to work with.

Normalizing the Y axis lets you view the full range of each curve (you view 0 to 100% of its Y value).
Displaying an Overview Region (Metacurve)

When you choose View > Metacurve Region, a bar below the graph shows a “dopesheet-like” view called a metacurve. This gives you an overview representation of all the keys on the selected fcurves in the graph. If the selection changes, it updates to reflect the new selection.

Viewing Corresponding Time with Action Sources

If you have created action sources from fcurves, you can see the corresponding time of the animation between the action in the animation mixer and the fcurve editor.

- You can select an action clip in the mixer and choose View > Local Time in the fcurve editor. You’ll see that the playback cursor in the fcurve editor shows the local time of where the original action source is being evaluated, and the playback cursor in the mixer shows where this point is in relation to the global time of the scene.

- You can also choose View > Driving Actions at Curr Frame in the animation tree to display parameters that are driven by an action clip in the animation mixer at the current frame (the playback cursor is at a frame that is covered by an action clip). Only the fcurves for the selected object’s current clip are displayed in the fcurve graph.

For more information on action sources and the fcurve editor, see Modifying Action Sources on page 123 in the Nonlinear Animation guide.
Zooming, Panning, and Framing in the Graph

You can use the Navigate, Zoom, Pan & Zoom, and Frame tools (available from the View menu, toolbar icons, and supra keys) to view a specific area of the graph.

To deactivate any viewing tool, press Esc or choose another tool.

By default, the navigation icons are not displayed in the toolbar. To display them, right-click in an empty area of the command bar and select Navigate (see Hiding Parts of the Command Bar on page 167).

Zooming

Interactively increase or decrease the magnification of the graph by choosing View > Interactive Zoom Tool, pressing c, or clicking the Zoom icon in the toolbar. The cursor turns into a small magnifying glass:

- To zoom in (increase magnification), drag in the graph to define a region to enlarge, or hold down the left mouse button.
  The region you draw defines the center of the region you want to see.

- To zoom out (decrease magnification), Shift+click and drag. Drawing a larger region zooms out less while drawing a smaller region zooms out more.

- To pan, middle-click and drag.

- To interactively increase and decrease the height of the view, right-click and drag up and down.
  To increase and decrease the width of the view, right-click and drag right and left.

- To reset the zoom factor, click once in the graph.

You can also zoom in and zoom out by choosing View > Pan and Zoom Tool, clicking the Pan icon in the toolbar, or pressing z.

- To zoom in, hold down the middle mouse button. The zoom-in continues until you release the mouse button.

- To zoom out, hold down the right mouse button. The zoom-out continues until you release the mouse button.

- To pan, drag with the left mouse button. The pan continues until you release the mouse button or the z key.
Chapter 7  •  Editing Function Curves

**Panning**

You can pan to scroll the area shown in the graph by choosing **View > Pan and Zoom Tool**, clicking the Pan icon in the toolbar, or pressing `z`.

- To pan, click+drag in the graph.
- To pan along the X axis, drag to the left or right.
- To pan along the Y axis, drag up or down.
- To pan across the graph during playback, choose **View > Automatic Pan**.

You can set the Pan icon to be the default “pan and zoom” tool by selecting it as the **Pan & Zoom Mode** on the View page in the Fcurve Editor Preferences property editor (see *Setting Fcurve Editor Preferences* on page 166).

**Panning and Zooming**

You can pan and zoom in the graph in any direction, including in constrained directions. You can use a number of different tools to pan and zoom in different ways.

**To interactively pan or zoom**

- Choose **View > Navigate Tool**, press `s`, or click the Navigate icon in the toolbar. Then:
  - To interactively pan the view, left-click and drag the mouse pointer in the graph.
  - To pan in a constrained direction, press Shift and left-click and drag in that direction.
  - To interactively zoom in, middle-click and drag to the left; to zoom out, middle-click and drag to the right.
  - To interactively zoom only horizontally, press Shift and middle-click and drag to the left to zoom in, and drag to the right to zoom out.
  - To interactively zoom only vertically, press Shift and middle-click and drag down to zoom in, and drag up to zoom out.

If you press Ctrl while this tool is active, you can switch to the rectangular zoom tool.

**To pan and zoom horizontally**

- Choose **View > Pan and Zoom in X Tool**, press `x`, or click the Horizontal pan & zoom icon in the toolbar.

**To pan and zoom vertically**

- Choose **View > Pan and Zoom in Y Tool** or click the Vertical pan & zoom icon in the toolbar.
Framing

Framing sets the zoom and pan so that the curves or keys fit in the graph within a specified range.

- To frame keys, select the keys and click the Frame Selection icon or press f. If no keys are selected, the selected curves or regions are framed.
- To automatically frame all fcurves in the graph when you open the animation editor with a selected object or update the selection, choose View > Auto Frame Curves. You can also set a preference for this in the Fcurve Editor Preferences property editor.
- To frame a region, highlight the region to be framed and click the Frame Selection icon, choose View > Frame > Selection or Region, or press f. If no region is selected, the selected curves are framed.
- To frame selected fcurves, click the Frame Selection icon, choose View > Frame > Selected Curves or Selection, or press f. If nothing is selected, then all curves are framed.
- To frame all displayed fcurves, choose View > Frame > All Curves, click the Frame Selection icon with nothing selected, or press a.
- To frame all the fcurves within the time span specified by the timeline, choose View > Frame > To Timeline or press r.
Chapter 7  Editing Function Curves

Editing Safely with Snapshots of Function Curves

By default when you begin editing an fcurve, a “snapshot” of the curve’s original shape is taken. This snapshot preserves a temporary copy of the original fcurve until you are satisfied with the new, edited version.

When you are satisfied with the new fcurve, you can take a snapshot of the newly edited curve and continue editing; otherwise, you can swap the new fcurve with its snapshot to undo your changes.

To replace the snapshot fcurve with a new snapshot

1. Make sure that Snapshot Curve is active (choose View > Snapshot Curve or make it active by default on the View page in the Fcurve Editor Preferences property editor—see Setting Fcurve Editor Preferences on page 166).

2. Edit an fcurve.

3. If you are satisfied with the edited curve, accept its new shape by clicking the Snapshot icon on the fcurve editor toolbar.

   A new snapshot is taken at the curve’s new position and the old snapshot is removed.

To swap an edited fcurve with its snapshot

1. Make sure that Snapshot Curve is active (choose View > Snapshot Curve).

2. Edit an fcurve.

3. If you want to return the fcurve to its original position, click the Swap with Snapshot icon on the fcurve editor toolbar. The edited fcurve and its snapshot exchange places.
Selecting Function Curves

When you select an fcurve, any modifications you perform are done to it. To select fcurves and their associated keys, select mode must be on. When selected, fcurves are highlighted in white and their keys are the color of the unselected fcurve (for example, Y parameters for transformations are green, so the unselected keys on the selected Y fcurve are green).

The selected fcurve is white with unselected keys matching the color of the unselected fcurve.

- If your fcurve has lots of keys on it (such as when importing motion capture information), use the animation tree to select a curve. This prevents keys on the other curves from being moved accidentally, and also lets you retain previously selected keys if you deselect the fcurve and then reselect it later.

- Choose View > Curve and Axis Labels to show or hide labels on all selected curves in the graph. Because fcurves for expressions and linked parameters do not necessarily use the same horizontal time units as other curves, you should select Curve and Axis Labels to know the meaning of their values.

To select both fcurves and keys

- Do one of the following:
  - Choose Edit > Select Tool.
  
or
  - Press y.
  
or
  - Click the Select icon in the fcurve editor’s toolbar.

To select only curves

- Do one of the following:
  - Choose Edit > Select Curve Tool.
  
or
  - Click the Select Curve icon in the fcurve editor’s toolbar.
  
or
  - Press the space bar.
  This tool selects only curves without changing the key selection.
To select and move fcurves

- Choose Edit > Select and Move Curves or press n.

To select a single fcurve

Make sure the appropriate select tool is active, then do any of the following:

- If the fcurve is visible in the fcurve editor, click it.
- If the fcurve is not visible in the fcurve editor, click the curve’s associated parameter in the animation tree.

To keep an fcurve selected while deselecting keys, middle-click+drag around the keys to deselect them or choose Edit > Deselect All Keys.

To select multiple fcurves

There are several ways to select multiple fcurves. First activate the appropriate select mode, then do one of the following:

- Shift+click the associated parameters in the animation tree or the parameters’ fcurves in the graph to add them to the existing selection.
- Ctrl+click the associated parameters in the animation tree or the parameters’ fcurves in the graph to toggle their selection.
- Drag the mouse pointer across a range of fcurves in the graph; the red selection box indicates the region you are covering. Any curves that pass through the box are selected.
- Choose Edit > Select All Curves from the fcurve editor’s command bar to select all curves currently displayed in the graph.

Deselecting Fcurves

To deselect fcurves

- Click in an empty area of the graph. This deselects all fcurves.

  If there were any selected keys on the fcurves before you deselected them, they are “remembered” when you select the fcurves again.

- Choose Edit > Deselect All Curves from the fcurve editor’s command bar to deselect all fcurves. This does not “remember” selected keys on the fcurves the next time you select them.
Selecting Function Curves

or

- Ctrl+click to toggle the selected fcurve.

If there were any selected keys on the fcurves before you deselected them, they are “remembered” when you select the fcurves again.

Setting the Default Selection Tool

By default, the Select tool is active, but you can choose which selection tool you want to be the preferred one.

To set a preference for the selection tool

1. Open the Fcurve Editor Preferences editor (see Setting Fcurve Editor Preferences on page 166).

2. On the Editor page, select the Default Tool:
   - Select Tool — selects both fcurves and keys.
   - Select Curve Tool — selects only fcurves.
   - Select Key Tool — selects only keys.
   - Select and Move Key Tool — selects and moves keys.

Changing the Selection Area around an Fcurve

By default, there is a buffer of 8 pixels around an fcurve in which you can click to select it. Normally, you don’t need to change the size of this area, but you can if you like.

To change the selection area around an fcurve

1. Make sure you have opened the animation editor (with the fcurve editor displayed) at least once in your XSI session.

2. Close XSI and restart it, but don’t open the animation editor yet.

3. In the script editor, enter and run the following lines (replacing 4 with the number of pixels you want):

   ```plaintext
   SetUserPref "AE Pick Size", 4
   LogMessage "New AE Pick Size = " & GetUserPref( "AE Pick Size" )
   ```

4. Now open the animation editor.

   The selection area around the fcurve should be the number of pixels you set. This new value is now kept as the default area.
Selecting (Tagging) Keys

You can select one or more keys on fcurves. Selected keys are highlighted in red and their exact coordinates are displayed on the X and Y axes of the graph.

In addition, keys using spline interpolation have slope handles to modifying the slope orientation and length (see Editing a Function Curve’s Slope on page 212 for more information).

- When you’re selecting keys, they have priority over other selected curves that are in the same location. However, if you are having trouble selecting a key because it is in the same location as a slope handle’s end point, make sure that Edit > Select with Priority to Tangents is not active.

- Use the View > Keys on Unselected Curves to show or hide keys on all unselected curves in the graph. If you choose this command, you can select keys without having to first select the fcurve. The default is to hide these keys, which prevents you from accidentally manipulating keys.

- Choose View > Keys Coordinates to show or hide the current time and value of a selected key.

To select both fcurves and keys

- Choose Edit > Select Tool or press y.

To select only keys

Do one of the following:

- Choose Edit > Select Key Tool.

  or

- Click the Select Key icon in the fcurve editor’s toolbar.

  or

- Press t.

This tool cannot select keys on unselected curves.

To select and move keys

- Choose Edit > Select and Move Keys.
Selecting (Tagging) Keys

To select one or more keys
Make sure the appropriate select tool is active, then do any of the following:

- To select a single key, click it.
  or
- To select multiple keys, Shift+click the points you want to select or drag a red rectangle around unselected points.

When you select a curve and then drag a red rectangle, only the keys on the selected curve are selected.

or

- To select all the keys on the currently selected curves, choose Edit > Select All Keys from the fcurve editor command bar.
  or
- To add or remove a key from the current selection, Ctrl+click the key point.

To move between keys on the selected fcurve

- Click the previous key (back arrow) or next key (forward arrow) icons on the command bar.
  or
- Press the , (comma) and . (period) keys.

Deselecting Keys

To deselect keys

- To deselect keys, click an empty spot in the graph. This also deselects all the fcurves. When you do this, any keys that were selected before are “remembered” when you select the fcurve again.
  or

- To deselect all keys on the currently selected curves, choose Edit > Deselect All Keys from the fcurve editor command bar. This deletes the keys but keeps the fcurves selected.
  or

- To deselect multiple keys while keeping the fcurve selected, middle-click+drag a rectangle around them.
Selecting Keys by Region

Using the region tool, you can select regions of points, making it easy to edit a large number of points at one time. You can edit a region by moving, stretching, or compressing all the keys within the region as a group.

To move or scale regions, see Moving Keys by Region on page 193 and Scaling Regions Interactively on page 204.

To select a region of points

1. Select the fcurves you wish to modify.
2. Click the Region icon in the toolbar or press q, then drag the mouse pointer horizontally across a section of the curves. This selects all points on the selected fcurves in the region.
   
   Hold down the Alt key while dragging to snap the left and right edges of the region to keyframes. The selected area is highlighted in light gray.

   If you want to extend or shrink the region you have selected, press Shift while you drag. You can also toggle curves and individual keys within the region by Ctrl+clicking them.

   Inside the region, between the maximum and minimum values of the selected keys and the first and last keys of the selected fcurve, is a highlighted selection area. This region is bounded by four handles.

   The region stays visible even if you change selection tools; however, it disappears when you click outside the region. As well, the last region you created for the fcurves is remembered the next time you select the same curves and activate the region tool.

   A handle bounds each side of the region. Drag these handles to resize the region.

To deselect a region of points

- Click anywhere outside the region.
Moving Function Curves, Keys, and Regions

When you move fcurves or keys in X, you change the keys' time; when you move in Y, you change the keys' values.

You can move fcurves, keys, and regions interactively or with numerical precision. You can also select a specific region of an fcurve in the graph and move, stretch, and compress the curve's keys within that region as a single unit.

You can randomize key values or apply expressions to keys using two Custom Fcurve tools in the Net View.

Moving Whatever is Selected

You can move either selected fcurves or keys using the Translate tool. It moves whatever is selected, with keys having the priority. For example, if fcurves are selected, it moves only those fcurves; if both fcurves and keys are selected, it moves only the keys.

To move selected fcurves or keys

1. Select fcurves or keys.
2. Choose Edit > Translate Tool, or press v, then drag the selected fcurves or keys to a new location.
   - To move only vertically, middle-click+drag.
   - To move only horizontally, right-click+drag.
   - To move multiple fcurves, first rectangle-select (Ctrl+drag) the fcurves you want, then press v and drag.

Moving Fcurves

To move fcurves interactively

• Choose Edit > Select and Move Curve Tool, or press n, then click+drag the curve to a new location.
   - To move only vertically, middle-click+drag.
   - To move only horizontally, right-click+drag.
   - To move multiple fcurves, first rectangle-select (Ctrl+drag) the fcurves you want, then press n and drag.

Moving Keys

When you move a key, you can change either the frame or the value of that key. Moving keys horizontally changes the frame, while moving vertically changes the value.

Because moving points interactively can be imprecise, you might want to have your keys snap to specific frames, time, or value intervals. For more information on snapping to the grid, see Snapping Keys on page 201.
To select and move keys

1. Choose Edit > Select and Move Keys.
2. Select the key points and drag them to a new location.

To snap keys while moving

- Press the Shift key while you drag selected keys to snap them to the Snap Grid coordinates that are specified in the Fcurve Editor Preferences property editor (see Snapping Keys on page 201).
- Middle-click+drag to move only vertically.
- Right-click+drag to move only horizontally.

To move keys without selecting them

You can move any key even while other keys are selected without changing the current key selection. This is similar to moving points on a geometric object in a view. You can move keys only on selected fcurves.

1. Choose Edit > Move Keys or press m.
2. Click a key point and drag it to a new location. The point turns green and has a yellow box around it as you’re dragging.

Only the point under the mouse pointer is affected, even if other keys are selected. The selection does not change.

To see the effect of the changes in the viewports while moving keys

1. Open the Fcurve Editor Preferences editor (see Setting Fcurve Editor Preferences on page 166).
2. On the View page, select Active in the Interactive Update section. You can adjust the Speed of the update by entering a value or dragging the slider.

At the top of the View page, you can also determine the level of control when moving keys by setting the Interaction Lag value. The default is the mouse moving one pixel before interaction starts.

To make sure that no keys are between frames after moving them, select them and choose Edit > Move Keys to Nearest Frame.
Moving Function Curves, Keys, and Regions

Moving Keys
by Region

To move keys by region

1. Select the region of the fcurves you wish to modify (click the Region icon in the toolbar or press q).

2. To change the frame position or values of the fcurve region, click anywhere within the highlighted region and drag the region:
   - Dragging the region horizontally adjusts the frame position of the region (the animation occurs earlier or later in the timeline). You can also right-click+drag to move the region only horizontally.
   - Dragging the region vertically changes the values of the entire region. All neighboring keys remain in their original positions. You can also middle-click+drag to move the region only vertically.

Moving Fcurves, Keys, and Regions Precisely

There are several ways to accurately move fcurves, keys, and regions to a specific location:

You can use:

- The Translate X (Tx) and Y (Ty) boxes.
- The Frame and Value boxes in the fcurve editor’s command bar.
- Set the values in the Key Properties editor.

To move fcurves, keys, and regions with the Translate boxes

1. Select the keys or region you want to offset. To offset the entire fcurve, select the curve and leave the keys unselected.

2. Enter the amount of the offset in the Tx or Ty text box, followed by a + (addition) or - (subtraction) sign.

   For example, to advance several keys by four frames, you would select the keys and enter 4+ in the Tx text box. To decrease the value by four points, you would select the keys and enter 4- in the Ty text box.

   You can also use functions such as linear (l) and random (r) in the Tx and Ty boxes. For more information, see Relative Input Using Math Operations on page 60 in the Interface and Tools guide.
To move fcurves, regions, or keys with the Frame and Value boxes

The Frame and Value text boxes show the frame and value of the currently selected key.

<table>
<thead>
<tr>
<th>Frame</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>74.33</td>
<td>153.57</td>
</tr>
</tbody>
</table>

1. Select the keys or region you want to offset. To offset the entire fcurve, select the curve and leave the keys unselected.

2. Enter the amount of the offset in the Frame or Value text box, followed by a + (addition) or – (subtraction) sign.

   For example, to advance several keys by two frames, you would select the keys and enter 2+ in the Frame text box. To decrease the value by two points, you would select the keys and enter 2- in the Value text box.

   For a single key, you can type in the exact frame number or value that you want. For example, if you know you want the selected key to have a value of 85, type this number in the Value text box—likewise for setting frames.

   You can also use the Frame and Value text boxes for scaling—see Scaling Function Curves, Keys, and Regions on page 203.

To change values in the Key Properties editor

You can also enter the new point coordinates for a series of neighboring keys in the Key Properties editor:

1. Select the key you want to move.

2. Choose Keys > Key Properties to open the Key Properties editor.

3. Enter the key coordinates in the Frame and Value text boxes.

4. Move to the next or previous keys by clicking the Previous Key and Next Key buttons and continue entering key values until you are done.

   In the Frame text box, you can specify the value in SMPTE time code format (that is, HH:MM:SS:FF) or you can enter the position in terms of number of frames (for example, at 30 fps, entering a value of 45 would be the same as entering a value of 00:00:01:15).
Adding and Deleting Keys

When you add or delete keys on an `fcurve`, it is the same as adding or deleting keyframes. XSI recalculates the interpolation and draws new curve segments reconnecting the keys.

If you add a point on the same frame as an existing key, the old point is replaced.

Adding Keys Interactively

To add keys interactively

1. Select the fcurves to which you want to add points.
2. Do one of the following:
   - Activate Add Key mode by clicking on the Add Keyframe icon in the toolbar or by pressing `i`. On the fcurve, click where you want to add keys.
   - or
   - Choose Keys > Edit Key Tool, click the Edit Key icon, or press `Shift+e`. Then middle-click to add keys.

Because adding points interactively can be imprecise, you might want to have your keys snap to specific time or value intervals. For more information on snapping to the grid, see Snapping Keys on page 201.

Inserting Keys without Changing the Fcurve Shape

You can insert a key anywhere on an `fcurve` without changing the shape of the `fcurve`. This is especially useful with curves that have few keys, or curves that have sharp peaks and valleys where adding a key as usual would change the shape of the `fcurve`.

1. Click the Add Keyframe icon in the toolbar or press `i`.
2. Hold down the middle-mouse button and drag the red vertical bar that appears. The small box on the bar follows the shape of the `fcurve` as you drag.

   Drag red box along the curve to where you want to add the key.

3. Release the middle mouse button at the place where you want to add the key. The key is added without changing the shape of the curve.

   Press Ctrl while dragging to snap to the increments set in the Fcurve Editor Preferences property editor (see Snapping Keys on page 201).
Inserting Keys at a Certain Frame

You can automatically add keys to all selected curves on a given frame. While a new key is added for each curve, the curves retain their shape. This is useful when you want to insert a key for a group of parameters (for example, X, Y, and Z) so that each key occurs at the exact same point in time.

**To add keys to multiple fcurves**

1. Select one or more fcurves to which you want to add keys.
2. Position the playback cursor to the frame where you want the keys inserted.
3. Choose **Keys > Insert Key at Current Time** from the fcurve editor command bar.

One key is inserted on each selected curve at the frame indicated by the playback cursor.

Inserting Keys at Key Time

As in the main XSI timeline, you can right-click the playback cursor in the animation editor’s timeline to go to a different frame without updating the scene. When you do this, the playback cursor turns green and you can set a key for these values.

**To copy and insert a key’s value**

1. Right-click in the animation editor’s timeline to set the key time. The playback cursor turns green.
2. Select the fcurve on which you want to insert the key and choose **Keys > Insert Key at Key Time**. This inserts a key on this fcurve without changing its shape.

If the key time is not visible (it’s at the same location as the current time), the command is dimmed since the current time is the key time.
Deleting Keys

When you delete keys from an fcurve, it is the same as deleting a keyframe from your animation. XSI recalculates the interpolation and draws a new curve segment connecting the remaining keys.

To delete specific keys using the Delete Key tool
1. Select the fcurves from which you want to delete points.
2. Choose Keys > Delete Key Tool, click the Delete key icon on the toolbar or press d.
3. Left-click the keys you want to delete.

To delete specific keys using the Edit Key tool
1. Choose Keys > Edit Key Tool, click the Edit Key icon or press Shift+e.
2. Right-click the keys you want to delete.

To delete multiple keys
1. Select the curve or curves from which you want to delete keys.
2. Select all the keys you want to delete.
3. Press Delete.

To delete multiple keys on the same frame
1. Select the curves from which you want to delete keys.
2. Move the playback cursor to the frame at which you want to delete keys.
3. Choose Keys > Delete Key at Current Time.

To delete a key at key time
1. Right-click in the animation editor’s timeline and drag the green cursor to where there is a key.
2. Select the curves from which you want to delete keys.
3. Choose Keys > Delete Key at Key Time.
Removing Animation from Fcurves

When you delete all the keys on an fcurve, you remove the animation of the associated property and the property’s fcurve reverts to a flat line, but remains in the graph.

For more information on removing animation in general, including removing fcurves from marked parameters, see Removing Animation on page 69.

To delete all keys on selected fcurves
1. Select the curves whose points you want to delete.
2. Choose Keys > Remove All Keys or press Backspace.

When you have removed keys from multiple fcurves, undoing (press Ctrl+z) only has an effect on the first curve you selected—the other curves remain flat-liners.

To delete fcurves
- Right-click the animation icon of an animated parameter in the animation tree and choose Remove Animation. This deletes all fcurves on that parameter completely; however, it also deletes any expressions or linked parameters set for that parameter.

If you mark parameters and then choose Remove Animation, all animation is removed for all marked parameters.

To delete flat (static) fcurves
- In the Animation panel at the bottom of the XSI window, choose Animation > Remove Animation > from All Parameters, Static Fcurves.

This removes fcurves that have no animation on them anymore (the fcurves are flat or static) from all parameters on selected objects.
You can flip an fcurve or region of keys on their X (time) or Y (values) axes, thus inverting the animation by time or its values. Flipping transposes the keys across a dividing line determined by the midpoint between the highest and lowest key values on the selected axis.

- Flipping an fcurve flips the entire curve, with the pivot centered on the curve.
- Flipping a region flips the fcurve segment defined in the whole region and not just the keys within it. The pivot is the center of the region. If you flip a region that does not start or end on a key, XSI creates new keys at the start or end point.

Flipping works by cutting the fcurve or region and then pasting it back. The first and last keys in a region are not affected by this, so to make sure that they are, select these Continuity pasting options in the Fcurve Editor Preferences property editor: Left Key: Use Right Value and Right Key: Use Left Value. See Setting Options for Pasting on page 208 for more information.

You cannot flip a curve or region that has keys locked in the same axis. For example, if a key is locked in time, Flip Y (Value) works but Flip X (Time) does not work.

To flip an fcurve or region

1. Select the fcurve or select a region of keys that you want to flip.
2. Choose Curves > Flip Y to invert the values; choose Flip X to invert the time.
Locking a Key’s Position

You can freeze a key’s values as well as its location in time by locking the key’s position on the graph’s X and Y axis.

If the fcurve uses spline interpolation to calculate the transition to the next key, you can also freeze the key’s speed of transition and influence on the fcurve via the fcurve editor’s slope handle locking controls. For more information, see Editing a Function Curve’s Slope on page 212.

To lock a key’s position on the graph

1. In the graph, select the key whose value or location in time you want to freeze.

2. Do one or both of the following:
   - Choose Keys > Lock in X (Time) to lock the key’s location in time on the X axis. The keys turn green.
   - Choose Keys > Lock in Y (Value) to lock the key’s value on the Y axis. The keys turn blue.

   If keys are locked in both X and Y, they turn light turquoise.

   The key is frozen in place on the graph and cannot move in the locked axis until you choose Keys > Lock in X (Time) or Keys > Lock in Y (Value) again. You can still manipulate the key slope handles to change the shape of the slope, unless they have also been locked (see Locking Slope Handles on page 221).
Snapping Keys

When you're moving or adding keys, you can make them snap to time and/or value coordinates on the fcurve graph grid's X/Y axes.

To set up the snapping coordinates

1. Open the Fcurve Editor Preferences editor (see Setting Fcurve Editor Preferences on page 166).

2. On the Editor page in the Snap Grid controls, set the X (Time) in frame increments and the Y (Value) in value increments for the grid.

If more than one key is selected and translated, only the translation amount is snapped to the grid. The keys that were on the grid remain on the grid and the others remain outside of it. For example, if you have set the X (Time) grid to 5, you would only be able to translate keys horizontally in increments of 5 frames.

3. Select the Snap option below the X and/or Y options, depending on the coordinates to which you want the keys to snap. For example, if you want keys to snap only to frames, select only the Snap option below X (Time).

Selecting a Snap option is the same as choosing the Edit > Snap to Grid command.

To snap keys to the grid

- Do one of the following to have keys snap to the grid’s X (time) and Y (value) coordinates as specified by the Snap Grid coordinates in the Preferences property editor (see above):
  - Press the Shift key while you drag selected keys.
  - Middle-click+drag to snap only vertically.
  - Right-click+drag to snap only horizontally.

  or

- Choose Edit > Snap to Grid from the fcurve editor command bar to activate the “snap” mode, then drag the selected keys.

Choosing this command is the same as selecting a Snap Grid > Snap option in the Fcurve Editor Preferences property editor.
Chapter 7 • Editing Function Curves

To snap keys to the nearest grid point

- Select keys or fcurves and choose Keys > Move Key to Nearest Grid Point.

The selected keys or the keys on the selected curves are snapped to their nearest relative grid point (time [X] and value [Y] intersection), based on the current grid settings.

If you have selected multiple keys and they would be snapped to the same grid time, no snapping occurs for the time, but it will occur for values.

To snap keys to frames

- From the fcurve editor command bar, choose Edit > Snap to Frame.

When you move keys, they snap only to whole frames (such as 1, 2, 3, etc.), not between frames (such as 1.45, 2.38, etc.).

- If you want your keys to automatically snap to frames while you are working, choose Edit > Auto Snap to Frames (Discrete Time). This merges all the keys within the same frame boundary.

To snap keys to the nearest frame

- Select keys and choose Keys > Move Key to Nearest Frame.

For example, if a key is at frame 12.3, it’s moved to frame 12; if it’s at frame 12.56, it’s moved to frame 13.
Scaling Function Curves, Keys, and Regions

You can change the timing of your animation by scaling fcurves, keys, or regions of them. When you decrease the length of the fcurve or region (X axis), you speed up the animation; conversely, increasing the length slows down the animation. You can also scale fcurves, keys, and regions on the Y axis, which changes the animation values (the pivot point is 0).

If you have fcurves with many keys, try scaling them using the HLE (high-level editing) tool’s Absolute Scaling method (see Shaping High-Density Fcurves on page 229).

Scaling Precisely

To scale precisely, you can use the Scale X (Sx) and Y (Sy) text boxes, or the Frame and Value text boxes, all found in the fcurve editor’s command bar.

To scale keys, fcurves, or regions

1. Select the keys, fcurves, or region you want to scale:
   - If you scale keys in time (X axis), the pivot point is the left-most key’s time on the curve.
   - If you scale fcurves, the pivot point is 0.

2. Enter the amount of the scaling in any of these text boxes:
   - To change the length of the fcurve or the position of the key, enter a value in the Sx box.

   You can also use functions such as linear (l) and random (r) in the Sx and Sy boxes. For more information, see Relative Input Using Math Operations on page 60 in the Interface and Tools guide.

   or

   - Enter a value followed by a * (multiplication) or / (division) sign in the Frame text box. For example, to double the length of an fcurve, enter 2*.

   or

   - To change the value of the fcurve or key, enter a value in the Sy box.

   or
- Enter a value followed by a * (multiplication) or / (division) sign in the Value text box. For example, to divide the value in half, enter 2/.

**Scaling Regions Interactively**

You can scale an fcurve region in time (horizontally) or by value (vertically). If the Ripple icon on the toolbar is activated, stretching the region horizontally shifts unselected keys in the graph down the timeline (to the right). If Ripple is off, the region cannot be stretched any farther than neighboring keys. For more information on Ripple, see Rippling on page 207.

To scale a region interactively

1. Select the region of the fcurves you wish to modify (click the Region icon in the toolbar or press q).
2. Do one of the following:
   - To stretch or compress the region using the opposite boundary as the pivot, left-click+drag a handle.
   - To stretch or compress the region symmetrically using the center as a pivot, middle-click+drag a handle.

Dragging horizontally shortens or extends the region over time.

Dragging vertically increases or decreases the range of values in the region.

For information on how to move regions, refer to Moving Keys by Region on page 193.
Scaling with a Pivot

You can stretch selected keys as you do with a region except that you can set a pivot point anywhere in the graph area. When you move your mouse, the keys are scaled according to where you clicked (the pivot point). For example, if the pivot point is directly between keys, you can scale in symmetry.

You can also select a single key and scale with a pivot, which is especially useful if you're scaling the values on an fcurve (scaling vertically). When you scale a key with a pivot point, you don't need to reorient the key's handles because they are correctly scaled with the key.

To scale with a pivot point

1. Select one or more keys in an area of the curve you want to scale.
2. Do one of the following:
   - Choose Edit > Stretch with Pivot Tool.
   - Press b.
   - Click the Stretch icon in the toolbar.
3. Click where you want the pivot point to be: red crosshairs and a “target” shows the spot. The pivot location is displayed on the status bar at the bottom of the XSI window.
4. Click+drag the mouse to scale the keys:
   - Left-click+drag to scale either horizontally or vertically. You can scale in only one direction at a time. The direction is determined by which way you initially move the mouse (horizontally or vertically).
   - Middle-click+drag to scale only vertically.
   - Right-click+drag to scale only horizontally.
Cutting, Copying, and Pasting in the Graph

You can copy and paste keys, a curve region, or entire curves. You can also set paste options to control how keys are pasted—whether they replace the selection or are added to it.

You can use the standard keyboard shortcuts for the cut, copy, and paste functions: Ctrl+c (Copy), Ctrl+x (Cut), and Ctrl+v (Paste).

Function Curves

To cut, copy, and paste an fcurve

1. Select the fcurve you want to copy.
2. Copy the curve to the clipboard by choosing Edit > Copy from the fcurve editor’s command bar or cut it by choosing Edit > Cut.
3. Select a parameter (or its fcurve) as the target to which you want to paste.
4. Paste from the clipboard by choosing Edit > Paste.

Remember that the way the fcurve is pasted depends on its settings in the Fcurve Editor Preferences property editor—see Setting Options for Pasting on page 208.

You can continue pasting the clipboard contents to other curves.

Keys

To cut, copy, and paste keys

1. Select the keys you want to copy.
2. Copy them to the clipboard by choosing Edit > Copy from the fcurve editor’s command bar or cut them by choosing Edit > Cut. A black line appears in the graph.
3. Select an fcurve or region as the target onto which you want to paste:
   - To paste the new keys starting at a specific frame on the fcurve, click in the graph to move the black line to the frame.
   - To paste the new keys in a specific region of the fcurve, select the region using the Region tool.
4. Paste from the clipboard by choosing Edit > Paste. Remember that the way the keys are pasted depends on its settings in the Fcurve Editor Preferences property editor—see Setting Options for Pasting on page 208.

If the target is a region, the clipboard contents are scaled to fit the number of frames in the selected area.

You can continue pasting the clipboard contents to other curves or regions.
Regions

When you copy a region, XSI can copy just the selected keys or the fcurve segment in the region itself. This is determined by the method used to select the region before copying or cutting.

Remember that activating the Ripple icon changes the way the region is pasted—see the following section for more information.

1. Do one of the following depending on what you want to cut/copy:
   - To cut or copy the keys within a region, use the Region tool (press q) or select a series of keys on the curve.
   - To cut or copy the fcurve segment in the region, use the Select tool and drag a rectangular selection over the curve.

2. Copy it to the clipboard by choosing Edit > Copy from the fcurve editor’s command bar or cut them by choosing Edit > Cut. A black line appears in the graph.

3. Select an fcurve or region as the target onto which you want to paste:
   - To paste the new region starting at a specific frame on the fcurve, click in the graph to move the black line to the frame.
   - To paste the new keys in a specific region of the fcurve, select the curve region using the Region tool.

4. Paste from the clipboard by choosing Edit > Paste. Remember that the way the keys are pasted depends on its settings in the Fcurve Editor Preferences property editor—see Setting Options for Pasting on page 208.

   If the target is a region, the clipboard contents are scaled to fit the number of frames in the selected area.

You can continue pasting the clipboard contents to other curves or regions.

Rippling

The Ripple icon on the fcurve editor’s toolbar provides a quick way to paste a region using the Insert option (see the next section).

When rippling is not active, you can move, scale, or paste a region without affecting the other keys on the fcurve. However, with rippling active, the region offsets other keys that are further along on the timeline (to the right). The rest of the fcurve is moved to the right of the inserted region, causing a ripple effect.
To activate ripple
Click the Ripple icon (it turns blue) or choose Edit > Ripple before you paste the curve region (or scale it).

Setting Options for Pasting

In the Fcurve Editor Preferences property editor, you can determine how the clipboard contents will be pasted.

To set the options for pasting
1. Open the Fcurve Editor Preferences editor (see Setting Fcurve Editor Preferences on page 166).
2. Click the Paste Options tab to select from the following Paste Options:
   - Insert—The region on the clipboard is added to the selected fcurve to the left of the current selection. The rest of the fcurve is moved to the right of the inserted region.
   - Replace—The region on the clipboard is overlaid on the equivalent area on the curve. When you select Replace, you can paste either absolute or relative values.
   - Absolute—The region on the clipboard completely overwrites the equivalent area on the curve.
   - Relative—The region on the clipboard is pasted as an offset from the first key in the pasted area.

The Continuity Options determine how the first and last keys of the pasted curve region are integrated into the curve.
   - Left Key: Use Average—The first key of the pasted region is the average of the values of the copied and original curves at that point.
   - Left Key: Use Left Value—The first key of the pasted region uses the value of the original curve at that point.
   - Left Key: Use Right Value—The first key of the pasted region uses the value of the copied curve at that point.
   - Right Key: Use Average—The last key of the pasted region is the average of the values of the copied and original curves at that point.
   - Right Key: Use Left Value—The last key of the pasted region uses the value of the copied curve at that point.
   - Right Key: Use Right Value—The last key of the pasted region uses the value of the original curve at that point.
Curve A's keys have been copied and will be pasted on curve B. Note the black bar on curve B, which is where curve A's keys will be pasted.

When using Insert, the new keys push the existing keys to the right.

When using Replace with the Absolute option, curve A's keys are placed on top of the original keys.

The black bar has been moved to a new location (indicated by the dashed line for clarity) before pasting. Curve A's keys have been pasted using Replace with the Relative option.

Since the Left Key continuity option is set to Use Left Value, the first key is set to curve B's value at that point. The rest of curve A's keys are pasted relative to the first key.

The exception is the last key; the Right Key continuity option is set to Use Right Value, so it is also set to curve B's value at that point.
Choosing a Function Curve Interpolation Type

Function curve interpolation is a way of estimating how animation progresses between keys. When you set two or more keys for a parameter, XSI determines the interpolation between them.

By default, fcurves use a spline interpolation which creates a smooth ease-in and ease-out transition at each key. However, if you want to change the nature of the animation, you can switch to constant or linear interpolation.

To set the interpolation type

1. Select the keys (or region) between which you want the interpolation. If you select an fcurve, the interpolation is applied to the entire curve.

2. Do either of the following:
   - Choose an interpolation option from the Curves menu on the fcurve editor’s command bar.
   - If you have selected keys, you can also choose Keys > Key Properties and select an Interpolation option in the Key Properties editor.

3. Select one of these interpolation options:
   - Spline Interpolation is the default. It uses a curved spline interpolation that accelerates and decelerates into (ease in) and out of (ease out) each keyframe, resulting in a smooth transition.
     The degree of acceleration and deceleration before and after the keyframe is determined by the slope handles associated with the key. For information on the slope, see Editing a Function Curve’s Slope on page 212.

   You can also click the Spline icon on the fcurve editor toolbar.

   - Linear Interpolation connects keys by straight line segments. The movement is characterized by constant speed with sudden changes at each key, creating a mechanical feel to the animation.
     This may be useful for animating lights and colors, where you don’t need the ease-in and ease-out that a spline interpolation offers.
Choosing a Function Curve Interpolation Type

You can also click the Linear icon on the fcurve editor toolbar.

- **Constant Interpolation** repeats the value of a key until the next one. The movement is characterized by sudden changes at keys and static positions between keys.

While you may not want to use this interpolation for character animation, it might be useful for animating a cut from one camera to another.

**Setting the Default Interpolation Type**

You can determine how movement will be calculated in an fcurve between keyframes by default. This setting is used when fcurves are created.

*To set the default interpolation type*

1. Choose File > Preferences from the main menu.
2. In the explorer that appears, select Animation to open the Animation Preferences property editor.
3. Select one of the **Interpolation for Inserted Keys** options:
   - **Constant**, **Linear**, and **Spline** are the same as described previously.
   - **Automatic** (From Neighboring Keys) ensures that the slopes for the keys are automatically calculated from the neighboring keys. This is the default option and it uses splines to ease in and out of each key.
Chapter 7 • Editing Function Curves

Editing a Function Curve’s Slope

The fcurve’s slope determines the rate of change in the animation. By modifying the slope, you can change the acceleration or deceleration in or out from a key, thus making the animation change rapidly or slowly, or even reversing it. The steeper the slope’s orientation, the faster the values change.

You can change the slope of any fcurve that uses spline interpolation by using the two handles (called slope handles) that extend out from a key. By modifying the handles’ length and direction, you can define the way the curve moves to and from keys.

By default, the slope handles are tangent to the curve at their key. This keeps the acceleration and deceleration smooth, but you can also “break” the slope at a certain point to have a sudden animation acceleration or deceleration, or change of direction altogether (see Breaking and Unifying the Slope Angle on page 217).

• Use the View > Slopes on Unselected Keys command to show or hide the slope handles of all unselected keys on a selected curve. If you choose this command, you can manipulate handles without having to first select the key. The default, however, is to hide these slope handles, which prevents you from accidentally manipulating them.

• Many of the slope commands that are available from the Keys menu are also available by Alt+right-clicking on a slope handle’s end point.

Changing the Handles’ Length and Angle

The length of the slope handles determines the weight of the key’s influence on the fcurve. The longer the handle, the more the key influences the connecting curve segment. By having control over the handle length, you can also draw more complex curve shapes with fewer keys.

The same fcurve, with short, medium, and long slope handles. Notice how the length of the handle changes the shape of the curve.
As you change the orientation and length of the slope handles, you can also set how the handles move: the left and right handles can be manipulated independently, together, or locked in place.

For more information, see Breaking and Unifying the Slope Angle on page 217 and Controlling the Length of the Slope Handles on page 218.

You can determine the level of control when moving slope handles by setting the Interaction Lag on the View page in the Fcurve Editor Preferences property editor (see Setting Fcurve Editor Preferences on page 166). The default is the mouse moving one pixel before interaction starts.

To change the slope handles interactively

1. Select one or more keys on an fcurve with spline interpolation—slope handles appear on each selected key.

2. Click the end of one slope handle and drag it to a new location, changing the length and/or angle of the handle. All the other slope handles move in the same way as this one.

You can use the mouse buttons as follows:

- Left-click+drag to change both the angle (orientation) and length of the handle.
- Middle-click+drag to change only the angle of the handle.
- Right-click+drag to change only the length of the handle.

If you are having trouble selecting a slope handle's end point because it is in the same location or too close to a key, choose Edit > Select with Priority to Tangents or press the p key to give priority to the end point.
To change the slope handles precisely

- Enter values for the slope handles’ length and angle in these text boxes in the command bar.

- The slope handle length is in fcurve graph units, and you can set up to one decimal place for the value (such as 1.2, rounding off the numbers).

- The slope handle angle value can be between -90 and 90 degrees, with up to two decimal places (such as 45.23). However, when you select the Express slopes as ratio as in SOFTIMAGE|3D option (see below), you can enter any value, not just one between -90 and 90.

The display of the slope handle in the fcurve graph may not look exactly like the angle that you enter depending on the amount you’ve zoomed in to the graph.

When you enter a value in the slope handle angle text boxes, the value is displayed as an angle. However, you can set the display to express the ratio as Y (value) over X (time) values (see below).

To switch the slope angle value display

1. Open the Fcurve Editor Preferences editor (see Setting Fcurve Editor Preferences on page 166).

2. Click the Keys tab and select Express slopes as ratio as in SOFTIMAGE|3D.

When this option is active, the slope angle text boxes display the slope values as a ratio of Y values over X values, as they are calculated in SOFTIMAGE|3D.

Setting the Slope’s Orientation (Angle)

The Keys menu on the fcurve editor command bar contains options that control how slope handles move as you manipulate them. Select one of the slope options as described in this section.

Slopes Based on Neighborhood Keys (Automatic)

The Keys > Automatic Slope command computes the orientation and length of a key’s slope based on the interpolation of the fcurve between the previous key and the next key (neighboring keys).
With this command selected, you may notice that when you move a handle after breaking the slope, the key’s opposite handle jumps slightly and the curve it controls moves a bit. This is because automatic slopes get recomputed, and are affected by the slope orientation being broken. One way to work around this is to deselect Automatic Slope before editing the slope.

Creating Break Points in the Slope

The Keys > Zero Slope Length command sets both the left and right handle lengths to zero, effectively creating a break point. When this is selected, you cannot change the handle length (or angle) since it is locked to zero.

Pointing at the Neighboring Keys

The Keys > Point at Neighboring Keys Slope Orientation command constrains the selected keys’ handles so that they point at the neighboring (previous and next) keys. This usually creates sharp break points in the slope.

The handles stay pointed at the neighboring keys even when you move the key. If you move the handle itself, the constraint is removed.

Creating Flat Slopes (Zero Orientation)

The Keys > Zero Slope Orientation command constrains keys to a slope of zero. The slope handles remain horizontal so that only the length can be modified.

This is useful to prevent the automatic interpolations from going above or below your keyframed values as they approach keys, such as keeping curves from changing when you don’t want them to.
Plateau Slope

The Keys > Plateau Slope command automatically sets the slope of a key to zero if its neighboring (previous or next) key has the same value or if that key is a local minimum or maximum.

- A local minimum is a key where both previous and next keys have larger values.
- A local maximum is a key where the previous and next keys have smaller values.

The shape of the curve is retained, but a linear segment is used between the two adjacent points with the same value. This allows the curve to hold key values, preventing the automatic interpolation from going above or below keyframed values that define extremes.

Using this type of slope can be useful for keeping objects in one place at keyframes where they are supposed to be still, and moving smoothly at keyframes where they’re supposed to move (like preventing the backsliding of feet between keyframes in a walk cycle and preventing jerky movements that would happen with a linear interpolation).

When Plateau Slope is active, it affects all points on a selected curve.

Setting the Default Slope

You can set the default slope mode for any keys that are inserted on an fcurve. This information is set in the Animation Preferences property editor and is saved in your preferences file so that it’s active for more than the current session in XSI.

To set the default slope

1. Choose File > Preferences from the main menu.
2. In the explorer that appears, select Animation or click its icon to open the Animation Preferences property editor.
3. Select one of the Slope Orientation Constraint options.

- See the previous pages for descriptions of Plateau and Zero Orientation.
- None means that the slope’s orientation is free to be modified by you, and/or it will get automatically calculated based on neighboring keys.
Breaking and Unifying the Slope Angle

When slopes are unified (default), any change you make to one side of the slope at a key is reflected on the other side. However, you can “break” a slope at a key so that you can change either side independently.

You may want to break the slope to create a sudden change in motion at a key, such as an object hitting a wall or a bouncing ball hitting the ground and then bouncing up again.

To control the orientation of the slope

- Select keys and do one of the following:
  - Choose the Keys > Unified Slope Orientation command.
  
  or

  - Press Shift+o.
  
  or

  - Click the Slope Orientation icon in the fcurve editor toolbar.
  
  or

  - Choose Keys > Key Properties to open the Key Properties editor and select either of the Slope Orientation options (Unified or Broken).

- When the Unified Slope Orientation command is active, it keeps the slope handles of the selected keys unified as you move them, keeping a constant angle between them. Unified slopes provide smooth, unbroken function-curve segments, as values ease into and out of keys.

- When the Unified Slope Orientation command is not active, you can break the slope to control the slope’s shape on either side of the key individually. Depending on the slopes you define, the transition in the rate of the parameter’s change at the keyframe can be as smooth or as abrupt as you like.

Unified Slope Orientation is active. Notice the smooth curves.

Unified Slope Orientation is not active. Notice the sharp breaks in the curve.
When the slope is broken, you can drag the slope handle in any direction. You can also middle-click+drag to move only the handle’s angle, or right-click+drag to move only the handle’s length.

**Mirroring the Slope’s Angle**

When the Keys > Unified Slope Orientation command is active, you can create a mirror constraint between a key’s tangent handles. This way, the moves you make to one key’s handle is mirrored to its other handle. This also works well when the Keys > Unified Slope Length command is active.

*To mirror the slope angle*

1. Choose the Keys > Mirrored Slope Orientation command.
2. Drag the handle of a selected key whose slope orientation is unified. The movement is mirrored on the other key’s handle.

**Controlling the Length of the Slope Handles**

You can set the length of the slope handles on both sides of the key to be equal or not.

*To control the length of the handles*

- Select keys and do one of the following:
  - Activate the Keys > Unified Slope Length option.

    *or*

    - Press Shift+1 (L).

    *or*

    - Click the Slope Length icon in the fcurve editor toolbar.

    *or*

    - Choose Keys > Key Properties to open the Key Properties editor and select either of the Slope Length options (Unified or Broken).

- If this option is active, both handles are kept the same length. This is the case even if the slope orientation is broken.
- If this option is not active (default), the length of each slope handle is independent.
Setting the Default Handle Length

You can determine the default slope handle length for when fcurves are created. This information is set in the Animation Preferences property editor and is saved in your preferences file so that it’s active for more than the current session in XSI.

To set the default slope handle length
1. Choose File > Preferences from the main menu.
2. In the explorer that appears, select Animation or click its icon to open the Animation Preferences property editor.
3. Select one of the Slope Length Constraint options:
   - None: The slope’s length is free to be modified by you, and/or it will get automatically calculated based on neighboring keys.
   - Zero Slope Length: Creates sharp points (break points) on the curve, allowing you to manipulate the handles independently.
   - Unified Length: Keeps the length for the slope handles equal on both sides of the key point.

Creating Fixed Length (SOFTIMAGE|3D) Slope Handles

You can have slope handles drawn with a fixed length, as is the case in SOFTIMAGE|3D. This length corresponds approximately to the horizontal length spanning five frames. You may need to keep slope handles the same length for a number of reasons, especially if you are frequently exchanging files between SOFTIMAGE|3D and XSI.

To convert fcurves to fixed length handles
1. Select the fcurves you want to convert.
2. Choose the Curves > SI|3D Curve command.
3. A warning message appears stating that the curves may be changed by doing this, and prompts you to continue or not.

If you continue, the handles of the selected curves change to be a fixed length and are locked. The shape of the curves may also change.
Chapter 7 • Editing Function Curves

The slope length text boxes on the command bar (see Changing the Handles’ Length and Angle on page 212) become blank and are inaccessible after you have chosen this command for the selected curve. If several curves are selected and only some of them are using this command, the text boxes are still inaccessible.

4. To return to XSI-style handles, deactivate the Curves > SI|3D Curve by choosing it again.

To set fixed length as the default slope type
1. Choose File > Preferences from the main menu.
2. In the explorer that appears, select Animation or click its icon to open the Animation Preferences property editor.
3. Select SI3D (Fixed Length Slopes) as the Curve Style. XSI (Free Length Slopes) is the default.

Snapping Slope Handles

You can snap slope handles in two different ways:

• Snap the handle’s length to a specific number of Softimage units (default is 1), from 0.01 to infinity.

• Snap the handle’s angle to a particular degree (45 is the default), from between 1 and 90.

To snap the slope handles in length or angle
1. Open the Fcurve Editor Preferences editor (see Setting Fcurve Editor Preferences on page 166).
2. On the Editor page, set the Snap Slope > Length in units and/or Angle in degrees.
3. Click in the appropriate Snap checkbox to activate Length and/or Angle.
4. Select one or more keys.
5. Choose Edit > Snap Slopes to activate slope snapping and then drag a slope handle.

or

• Press the Shift key while you drag a slope handle to snap it in the increments you set. This temporarily activates the Snap Slopes command.
  - Middle-click+drag to snap only the handle’s angle.
  - Right-click+drag to snap only the handle’s length.
Locking Slope Handles

If the fcurve uses a spline interpolation, you can freeze a key’s speed of transition and influence on the fcurve by locking the orientation and length of its slope handles:

1. On an fcurve, select the keys whose handles you want to freeze.

2. Do one or both of the following:
   - Choose Keys > Lock Slope Orientation to lock the slope handles’ orientation. The slope handles turn blue.
   - Choose Keys > Lock Slope Length to lock the length of each slope handle. The slope handles turn green.

   The slope handles of the selected key are frozen in place and cannot be moved. You can still manipulate the position of the key itself, unless its position has also been locked (see Locking a Key’s Position on page 200).

3. Choose the same commands again to unlock the keys or slope handles.

To keep a fixed length for the handles, see Creating Fixed Length (SOFTIMAGE|3D) Slope Handles on page 219.
Extrapolating Function Curves

The extrapolation controls let you modify the shape of an fcurve before the first key and after the last key. Extrapolation extends the curve's ends based on the position of these two extreme points.

To apply extrapolation to a curve

1. Select an fcurve.
2. Choose Curves > Constant Extrapolation or Curves > Gradient Extrapolation from the fcurve editor command bar.

For spline interpolation, the slope of the first or last key is used. This results in a continuation of the motion at the same speed.

- **Constant** extrapolation flattens the curve after the end points, adding no animation. It extrapolates the value of the Y axis from the first frame to the first key, and the value of the Y axis from the last key to the last frame. This results in a hold of motion before and after the movement.

  With constant extrapolation, the values before the first keyframe and the values after the last keyframe remain constant.

- **Gradient** extrapolation makes the curve continue in the direction of (tangent to) the end keys’ slope handle angle. It extrapolates an increment based on the first key's slope orientation and applies it from the first frame to the first key. Then it extrapolates the increment calculated between the last and previous-to-last keys and applies it from the last key frame to the last frame. This has the effect of the curve continuing on this slope indefinitely.

  With gradient extrapolation, the values before the first keyframe and the values after the last keyframe are determined by the slope. Note the slope handles’ relation is tangent to the extrapolated values.
Creating Function Curve Cycles

The Cycle commands in the Curve menu on the fcurve editor’s command bar provide you with several quick ways to repeat animation. By using these controls, you can create repetitive motion including bouncing, spinning, or oscillation effects. You can create basic endless cycles or you can have relative cycles that are progressively offset (such as for creating a walk cycle).

When you create cycles, only the original fcurve retains its keys: the resulting cycles don’t have any. However, there is a “relational” link between the original keys and their counterparts on the resulting curves. This means that you can change a key in the original segment and those changes are automatically reflected in the cycles (you can add or delete points, move points, change the slope, etc.—it’s all updated in the cycles). You can also switch between different types of cycles at any time.

If you want the resulting cycles to have keys, you have to freeze the cycles as described in Freezing Fcurve Cycles on page 225.

Tips for Creating Seamless Cycles

• If the first and last keys points on the fcurve segment you want to repeat do not have the same value, there will be some jumping in the animation as the object snaps from the last frame to the first one of the next cycle. You can make seamless cycles by setting the first and last key to precisely the same value.

To do this, enter the same values for the keys in the Value box in the command bar or right-click either the first or last keyframe in the timeline and drag to the other end keyframe to copy the values exactly.

• If you have one or more keys on your fcurve that are not on a frame, this can produce unpredictable effects because the cycle is evaluated at each frame and not on an intermediate time. Choose Edit > Move Keys to Nearest Frames to remedy this.

Creating a Basic Cycle

The Curves > Cycle command repeats the fcurve shape as defined by its keys. This means that the part of the fcurve as defined by its first and last keys is cycled endlessly in both directions in time (X axis).

To create a basic cycle

1. Select one or more fcurves whose shape you want to repeat (that is, the shape between the first and last keys).

2. Choose Curves > Cycle from the fcurve editor’s command bar.
3. The fcurve segment is repeated continuously in both directions in time.

Creating a Relative Cycle (Offset)

The Curves > Relative Cycle command repeats the fcurve shape (defined by its first and last keys) by making each cycle start with the same value as the last key of the previous value. This results in a curve cycle with a progressive offset.

Setting a progressive offset is particularly important when you want to create a simple walk cycle. For example, animating a foot translation in a repeated cycle on the Y axis generally requires no offset, but cycling animation on the X axis without an offset would make the character look like he's walking without moving forward.

To create a relative cycle
1. Select one or more fcurves whose shape you want to repeat.
2. Choose Curves > Relative Cycle from the fcurve editor’s command bar.

Deleting Fcurve Cycles

Since a cycle's repeated information is not actually added to the fcurve, you can delete the cycle completely while leaving the original keyframes intact.

To delete the cycle information
- Select one or more cycled fcurves and choose Curves > Constant Extrapolation or Gradient Extrapolation.

For more information on extrapolation, see Extrapolating Function Curves on page 222.
Freezing Fcurve Cycles

The Curves > Freeze Cycle command freezes the selected fcurve by creating (or sampling) the keys represented on the cycled part of the fcurve. The cycles can then be edited as any fcurve because they now have key points.

To freeze an fcurve cycle

1. Select one or more cycled fcurves whose shape you want to freeze.
2. Do either of the following:
   - Drag the mouse pointer over the region of the fcurve you wish to freeze. Then choose Curves > Freeze Cycle from the fcurve editor command bar.
   - Choose Curves > Freeze Cycle. If no keys on the curve are selected and no region is highlighted, a dialog box appears in which you can enter the frame range in which you want the cycle to be frozen, including negative frame numbers.

The cycle is frozen with new keys on the cycled parts of the fcurve. You can now modify this fcurve as you would any other.

The original keys are indicated between the dashed lines. The frozen curve now has keys, which can be modified.
Editing Quaternion Function Curves

You can edit the keys on a quaternion fcurve the same as any other fcurve (such as moving, adding, deleting, copying/cutting and pasting keys) and the interpolation between the keys will be quaternion. By default, quaternions interpolate the shortest path between two rotations, so quaternion keys should be less than 180 degrees apart for best results.

You can control the keys on quaternion interpolation fcurves using four parameters (tension, continuity, bias, and spin), which are located in the Quaternion Keys toolbar. These parameters control the incoming and outgoing tangent vectors at each key, much like slopes on a regular fcurve.

Because quaternion rotation calculates orientation as a single entity and not individual rotation axis fcurves, any change you make to one rotation fcurve affects the two other fcurves. For example, if you add a key to the rotation X fcurve, a key is also added at the same frame for the rotation Y and Z fcurves.

For information on quaternion rotation in general, see Animating Rotations on page 153.

To edit quaternion fcurves

1. Select the object with quaternion fcurves and open the Fcurve Editor in the animation editor.

2. To edit quaternion fcurves, you need to use the Quaternion Keys controls, which are hidden by default.

   To display them, right-click in an empty area in the fcurve editor’s command bar and choose Quaternion Keys.

3. Select one or more keys on the fcurves and edit the slopes using the Tension, Continuity, Bias, and Spin controls.

   When you add, delete, or move a keyframe on one rotation fcurve, the corresponding key is automatically updated on the other rotation fcurves.
Editing Quaternion Function Curves

- When you type values in the quaternion parameter text boxes, the range is limited to values between -1 and 1. However, you can surpass this range by using the [ and ] bracket keys (also in conjunction with Shift and Ctrl) to modify the values by different increments. See Setting Values for Parameters on page 56 in the Interface and Tools guide for more information.

- Regardless of whether the rotation is quaternion or Euler, the Value box on the fcurve editor's command bar always displays Euler angle values.

- **Tension** controls the tightness of the curve or how sharply it bends at a key. Negative values produce a looser curve while positive values produce a tighter curve. Tension also has an effect on the speed: negative values causes the object to speed up (ease out) while positive values cause the object to slow down (ease in) as it passes through the key.

  ![Tension with a value of -3. Original fcurve is ghosted.](image1)
  ![Tension with a value of 2.](image2)

- **Continuity** controls how smoothly the curve passes through a key. Negative continuity gives a sharper more linear transition through the key while positive continuity gives a broader transition, bending the tangents in the opposite direction. Setting the continuity to -1 gives the same effect as quaternion spherical linear interpolation.

  ![Continuity with a value of -1. Original fcurve is ghosted.](image3)
  ![Continuity with a value of 1.](image4)
- **Bias** controls the direction of the curve through the key. A negative bias skews the curve towards the previous keyframe (undershoot) while a positive bias skews the curve towards the next key (overshoot).

- **Spin** controls the direction and number of revolutions between keys. You can use negative numbers to reverse the interpolation direction (that is, -1 can be used to flip the interpolation direction).

When using a non-zero spin value, the quaternion fcurves displayed in the fcurve editor will have discontinuities. This is only a display artifact and does not affect the interpolation.
Shaping High-Density Fcurves

A high-density fcurve is a curve with many keys, often one per frame, as you may get with motion capture or plotted data. This type of fcurve is often difficult to edit because if you change a few keys, you have to adjust many other keys to retain the overall shape of the curve. For example, moving a key results in a peak in the curve (as shown below):

To the rescue is the high-level editing (HLE) tool! This tool lets you shape an fcurve in an overall fashion, similar in concept to how lattices let you shape the geometry of an object. The HLE tool creates a “sculpting” curve that has few keys, but each one refers to a group of points on the dense fcurve. You then edit this curve, which shapes the dense fcurve using a simple arithmetic operation as defined by one of the three HLE operations.

To edit fcurves in HLE mode

1. Select one or more fcurves.

2. Click the HLE button in the fcurve editor’s command bar to activate the high-level editing mode.
Chapter 7 • Editing Function Curves

An HLE curve is created to let you shape the selected fcurves. It has keys on it as a regular fcurve so that you can edit it. The HLE curve exists and is viewable only when you’re in HLE mode. The HLE curve shown below is using the Relative Offset operation.

The curves you selected become the “slave” curves to the HLE curve and turn a pale blue: they cannot be selected in this mode, but you can see how the changes made to the HLE curve affect the “slave” curve.

To make it easier to see what’s happening, any fcurve that wasn’t selected is not visible in this mode.

3. Select an HLE operation from the list beside the C button (see next section for more information): Relative Offset, Absolute Offset, or Absolute Scale. You can also switch operations during an HLE session.

The Curve and Axis Labels are activated so that you can see which HLE operation is active.

4. Edit the HLE curve as you would any other fcurve (move, scale, add or delete points, etc.).

When you edit the HLE curve, the changes are transferred to its corresponding dense fcurve. You cannot edit the slave fcurve when in HLE mode.

5. To move the HLE curves without changing the slave curves, click the C button. Click this button again to have your edits affect the slave curves.

6. When you’re finished editing, click the HLE button to deactivate HLE mode.

The slave curve is updated with the new shape, and becomes selectable again. All other fcurves are also displayed again.

Although you cannot reset an HLE curve, you can take a snapshot of it before you modify it, then swap with the snapshot if you need to (see Editing Safely with Snapshots of Function Curves on page 184).

You can also undo the operations on the HLE curve (press Ctrl+z).
Selecting an HLE Editing Operation

Only the HLE curves corresponding to the current operation are visible; if you change the operation while in HLE mode, you’ll see new HLE curves appearing for this operation.

- **Relative Offset** creates one HLE curve per slave curve. Each of these HLE curves is obtained by low-pass filtering (smoothing) the corresponding slave and resampling the result (so there are fewer keys) before locking the keys in time. The ratio of keys is 10:1 (10 keys on fcurve related to 1 key on the HLE curve).

When editing the HLE curve, the difference between the default curve and the current curve is added to the slave curve.

- **Absolute Offset** creates one HLE curve for all the slave curves. This HLE curve has 2 keys with a value of 0 at frames that correspond to the first and last keys of all slave curves. When editing this HLE curve, the difference between the default curve and the current curve is added to all slaves.

- **Absolute Scaling** does the same as Absolute Offset except that the slaves are scaled by the ratio current curve over the initial curve. Initial curve keys have a value of 1.
With Absolute Scaling, updating the reference HLE curve can cause problems when some of its values get close to 0.
**Processing Function Curves**

Function curves created within XSI usually have keys that are located at various frames throughout your animation. XSI evaluates the fcurve and interpolates between the keys.

Fcurves generated from external sources such as motion capture may have many more key frames than necessary (commonly one key per frame) or they may contain “noise”—sharp spikes and jags in the fcurve. You may want to remove some of the keys or reduce the noise to edit the animation.

Conversely, if you keyframe an animation in XSI and you want to transfer the animation data to an external device that requires frame-by-frame information (for example, a camera), you may want to increase the number of keyframes and preserve the motion as best as possible.

**Reducing Noise (Smoothing)**

Smoothing decreases the variation between consecutive keyframes on the fcurve. It recreates keyframes at a regular time interval along the fcurve. It also modifies each keyframe’s value, resetting it closer to the average value of all keyframes so that the result of a repeated application of the Smooth command would be a flat curve!

**To reduce noise on an fcurve**

1. Select one or more fcurves that you want to smooth.
2. Choose Curves > Curve Processing Options or open the Fcurve Editor Preferences editor (see Setting Fcurve Editor Preferences on page 166).
3. On the Curve Processing page, choose the **Smoothing** method:

<table>
<thead>
<tr>
<th>Smoothing</th>
<th>Average Filter</th>
<th>Gaussian Filter</th>
</tr>
</thead>
<tbody>
<tr>
<td>Filter Size</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Variance</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>Fitting Tolerance</td>
<td>0.1</td>
<td></td>
</tr>
<tr>
<td>Resample Before Fitting</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Resampling Time Step</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>Keep Existing Keys</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Use **Average Filter** to apply a moving average where the weights of the averages are uniformly distributed. You can define the number of keys to average at a time using **Filter Size**. A greater number of keys results in a straighter curve.
Chapter 7 • Editing Function Curves

- Use **Gaussian Filter** to apply a moving average where the weights of the averages are distributed as a bell curve. Use the **Variance** setting to control the degree of smoothing. A higher variance results in a smoother curve. The Gaussian filter usually provides better results than the **Average** filter.

4. Choose **Curves > Smooth** from the fcurve editor command bar.

![Average filter used with a value of 4. Original fcurve is shown in black.](image1)

![Gaussian filter used with a value of 4. Original fcurve is shown in black.](image2)

**Decreasing the Number of Keyframes (Fitting)**

Fitting reduces the number of keys on the fcurve while preserving the original shape of an fcurve. You can also use the **Resampling** option to reduce the number of keys (see the next section).

*To reduce the number of keyframes on an fcurve*

1. Select one or more fcves that you want to fit.

2. Choose **Curves > Curve Processing Options** or open the Fcurve Editor Preferences editor (see Setting Fcurve Editor Preferences on page 166).

3. On the Curve Processing page, define a **Fitting > Tolerance** value to set the processing accuracy.

   The higher the value, the more points are removed resulting in the curve’s shape deviating more from the original. Smaller values yield a closer fit to the original curve but can create more keys.

4. Choose **Curves > Fit** from the fcurve editor command bar.

![Original motion capture fcurve with one key per frame.](image3)

![Fcurve fitted with a Fitting Tolerance value of 1.](image4)
Resampling before Fitting

If you are using a low-density fcurve (one with few keys), you may find it useful to resample a curve before fitting. This way, you can apply a fit using a spline curve to help preserve the smoothness of the resulting fitted curve. If you fit a low-density curve without doing this, the segments between the keys may be more linear than you want.

To do this, select **Resample before fit** and specify a step by which to resample. The resampling is performed first, then the fitting action as previously described.

Resampling adds or removes keyframes at the interval that you specify.

To change the number of keyframes on an fcurve

1. Select one or more fcurves that you want to resample.
2. Choose **Curves > Curve Processing Options** or open the Fcurve Editor Preferences editor (see Setting Fcurve Editor Preferences on page 166).
3. On the Curve Processing page, set the **Resampling** > **Time Step**. The time step determines the frequency of the generated keyframes.
   - For instance, a time step of 1 creates a keyframe on every frame, and a time step of 2 creates a keyframe on every second frame.
   - You can also choose to keep or discard your original keys by selecting **Keep Existing Keys**.
4. Choose **Curves > Resample** from the fcurve editor command bar.

Original motion capture fcurve with one key per frame.

Fcurve resampled with a **Time Step** value of 2 to have one key at every second frame.
Deactivating (Muting) Function Curves

You can deactivate (mute) a function curve so that its animation has no effect on the object. Muting a function curve doesn’t remove it from the object, it just temporarily disables it.

You can also mute function curves in the dopesheet, including regions of the function curves. For more information, see Deactivating (Muting) Animation in the Dopesheet on page 261.

Muting and unmuting an fcurve cannot be undone or redone using the Undo or Redo commands.

To mute function curves
1. Select one or more function curves.
2. Choose Curves > Deactivate Curve from the fcurve editor command bar.

The curves appear as dotted lines. When you deselect them, they are dotted lines in their usual color (such as blue for Z-axis transformation fcurves).

You can select the muted curves and edit them as you would normally.

To activate muted function curves
1. Select one or more muted function curves.
2. Choose Curves > Activate Curve.
Converting Between Raw and Standard Fcurves

You can convert from raw to standard spline fcurves, and vice versa, in the fcurve editor.

Converting from raw to standard fcurves is useful for saving memory and increasing performance when using fcurves from motion capture animation or plotted animation.

- Standard fcurves created have fewer points and have spline interpolation.
- Raw fcurves created from plotted actions are not supported.

When you convert a standard spline fcurve to raw, the raw format includes only the minimal information needed (no slopes and the interpolation is linear).

To convert from raw to standard fcurves
1. Select one or more raw fcurves.
2. Choose the Curves > Convert to Standard command.
3. You are prompted with a warning that “This command is not undoable. Do you want to proceed?” This process is not undoable to save memory.

   Click Yes to convert the curve to a Standard format.

To convert from standard to raw fcurves
1. Select one or more standard (spline) fcurves.
2. Choose the Curves > Convert to Raw command.
3. You are prompted with a warning that “This command is not undoable. Do you want to proceed?” This process is not undoable to save memory.

   Click Yes to convert the curve to a Raw format.
Applying Preset Function Curves

There are a number of presets available for time-independent fcurves that allow you to quickly apply a certain shape. These are fcurves that do not represent change over time, but rather some kind of relationship with another parameter, such as for deform profile curves, linked parameters, and expressions.

These presets are also available in “mini” fcurve editors as found for many of the deform operators (see Deformations on page 137 in the Modeling and Deformation Basics guide) and for the time warp function in the animation mixer (see Changing Time Relationships (Timewarps) on page 75 in the Nonlinear Animation guide).

Presets are fcurves saved in .fraw2 ASCII files (see Saving and Loading Function Curves on page 239). By default, their range in X (normally the “time” axis) is from 1 to 100, and 0 to 1 in Y (value).

To apply a preset to a time-independent fcurve

1. Select a time-independent fcurve (you can select only one at a time).
2. Right-click in any empty space in the fcurve graph.
3. Choose one of the several presets from the menu (as shown on the left).

The preset fcurve is remapped (scaled) in X to match the first and last key of the selected fcurve. If there are fewer than two keys, than the preset is loaded with the default X and Y ranges.
Saving and Loading Function Curves

The fcurses you create in the fcurve editor can be saved as .fraw2 files and subsequently applied to other parameters. You can also import and export SOFTIMAGE|3D fcurses as .fraw files.

The .fraw format of the SOFTIMAGE|3D raw fcurve file is an ASCII file with a minimal description of fcurses as follows:

<number of keys> <interpolation mode: 0 = CONSTANT, 1 = LINEAR, 2 = SPLINE>

The .fraw2 format is a simple ASCII file that can be used to import and export fcurses to and from XSI. It is an augmented version of the .fraw format that can describe any fcurve in XSI. The file can contain several fcurses, each of which is described following this format:

Start_FCurve <curve name> <number of keys> <curve type> <extrapolation method> <value when no keys> <min clamping value> <max clamping value> <first segment kind> <unused field (for padding)>

Because this file format is ASCII, it's simple to write a converter for other software packages. It also makes it easy for you to save fcurve presets.

For more details on the .fraw2 format, see the online help in the animation editor.

Saving Fcurses

To save an fcurve preset in an .fraw2 file

1. Select the fcurve you want to save.

   If you select multiple fcurses, the fcurses are saved in the .fraw2 file, but only the first one is loaded. This is because the file contains only raw data—there is no mapping information.

2. Choose File > Save from the fcurve editor command bar or press Ctrl+s (with the mouse pointer within the fcurve editor) to save the fcurve in an .fraw2 file for use in XSI.

To save an fcurve for exporting to SOFTIMAGE|3D in a .fraw file

1. Select the fcurve you want to save.

2. Choose File > Export (.fraw) to export the fcurve for use in SOFTIMAGE|3D.

3. In the browser, navigate to the folder where you want to save your curve, give it a name, and click OK.

   Your curve is now saved as an fcurve file, indicated by its .fraw2 (XSI) or .fraw (SOFTIMAGE|3D) extension.
Chapter 7 • Editing Function Curves

**Loading Fcurves**

*To load an fcurve in the fcurve editor*

1. In the fcurve editor, select the fcurve you want to replace.
2. Do one of the following:
   - Choose **File > Open** from the fcurve editor command bar or press **Ctrl+o** to load an .fraw2 fcurve.
   
   or
   
   - Choose **File > Import (.fraw)** to import an .fraw fcurve from SOFTIMAGE|3D.

The loaded fcurve replaces the currently selected curve.
Chapter 8  Editing Animation in the Dopesheet

The dopesheet provides you with a way of viewing and editing keyed animation. Similar to a cel animator's dopesheet, it shows your entire animated sequence, frame by frame. Because you can see your whole animation, you can accurately judge its overall motion and timing, allowing you to block out the animation. Keyframes are easily identified on the tracks as colored boxes.

The dopesheet gives you a good overview of your animation so that you can carry out large animation tasks, such as retiming all or just segments of an animation, with a minimum amount of effort. For example, if you wanted to change a 100-frame sequence to 120 frames, you would stretch (scale) the animation segment on the track to be 120 frames long.

You can also use the Sequence commands in the Animation menu to scale and offset animation globally. For information, see Scaling and Offseting (Retiming) Animation on page 55.
You can modify your animation sequences by editing regions of keys on the tracks with standard operations such as moving, scaling, copying, cutting, and pasting. You can delete them, shift them left and right, scale them—all with or without a ripple.

The dopesheet is part of the animation editor, so it shares the same animation tree and timeline. However, the dopesheet has its own command bar which contains tools that manipulate the keys of the selected element’s parameters.

To display the dopesheet

There are three ways:

- Choose Dopesheet from a viewport’s view menu. This opens up the animation editor with the dopesheet as the active editor.

  or

- In the animation editor, choose Editor > Dopesheet from its command bar.

  or
• To display the dopesheet (within the animation editor) in a floating window, choose View > Animation > Dopesheet from the main menu bar. You can move and resize this like any other window.

To display the dopesheet in OpenGL instead of GDI, make sure that Edit > Use OpenGL is active. You can set a default for this in the dopesheet preferences property editor (see Setting Preferences for the Dopesheet on page 243).

“Dopesheet” in the Timeline

Similar to the dopesheet, you can display and edit keys in the timeline for the selected object. This can be handy for viewing and editing keys when you are working in a full-size viewport and can’t have a full dopesheet open.

You can edit regions of keys in the timeline as you do in the dopesheet. For more information, see Editing Keys in the Timeline on page 125.

Setting Preferences for the Dopesheet

The dopesheet preferences let you set up options for the dopesheet displayed within the animation editor, as well as the keys displayed in the timeline (see Setting Preferences for Keys in the Timeline on page 138).

There are also preference property editors for the fcurve editor (see Setting Fcurve Editor Preferences on page 166) and the animation tree in the animation editor (see Setting Preferences for the Animation Tree on page 163).

To set the dopesheet preferences

1. Do either of the following:
   - Choose Edit > Preferences in the dopesheet command bar.
   
   or
   
   - Choose File > Preferences from the XSI main menu.

2. In the explorer that appears, select Editors > Dopesheet or click its icon.

3. Set the preferences for the dopesheet, such as display options, paste operations, phoneme keys, and many others.
Controlling What Is Displayed in the Dopesheet

The dopesheet responds to the currently selected elements. While it is open, it displays the animation (if any) of the elements you select.

In the animation editor’s tree, you can select and control the type of parameters that are displayed in the dopesheet. This tree is essentially the same as the explorer and is used to control what is shown in the animation editor, regardless of which editor is currently active (dopesheet, function curves, expression, or scripted operators). For information on using this, see Using the Animation Tree on page 162.

Once you have the parameters you want shown in the animation tree, you can control the display of their tracks with the commands in the View menu that is above the tracks:

- View > Animated Parameters shows the tracks of all animated parameters for the selected object. This is selected by default.
- View > Marked Parameters shows only the tracks of marked parameters for the selected object.

View > Marked Parameters does not apply to marking sets, only to parameters that are actually marked (see Marking Parameters on page 36 to know which parameters are marked). Marking sets are actually custom parameter sets.

To see marking sets, choose View > Custom Properties and make sure that View > Animated Parameters Only is off for the animation tree on the left. You can also choose the flattest display mode, View > Flatten All. Now in the tree, you will see the marking set with its parameters beneath it.

- View > Selected Parameters shows only the tracks of selected parameters for the selected object.

You can set a default for which View command is active in the dopesheet preferences property editor (see Setting Preferences for the Dopesheet on page 243).

To display tracks for unanimated parameters

1. Make sure that View > Animated Parameters Only is off for the animation tree.
2. Choose View > Selected Parameters.
3. In the animation tree, browse for and select the unanimated parameters for which you want a track.

Selecting a folder (such as kine.local) generates tracks for all unanimated parameters within it.

4. An empty track appears for each unanimated parameter you select.
Zooming, Panning, and Framing in the Dopesheet

You can use the Navigate, Zoom, Pan, and Frame tools (available from the Tools menu), or their corresponding shortcut keys to view a specific area of the dopesheet.

- To reset the view of the dopesheet so that the tracks are framed within the window, choose View > Frame to Timeline or press r.
- To deactivate any viewing mode, choose another Tools command or press Esc.

Zooming

You can interactively increase or decrease the magnification of the dopesheet view using the zoom tool or the pan & zoom tools.

To zoom

- Choose Tools > Rectangular Zoom Tool or press Shift+z, then:
  - To rectangle-zoom in (increase magnification), click+drag in the dopesheet to select a rectangular region to enlarge.
  - To rectangle-zoom out (decrease magnification), right-click and drag to draw a region. The region you draw defines the center of the region you want to see; you define a large region to zoom out less and a small region to zoom out more.
  - To zoom interactively, hold down the middle-mouse button and drag.

Panning and Zooming

You can pan to scroll the area shown in the dopesheet, even specifically in the horizontal and vertical directions. You can use a number of different tools to pan and zoom.

To pan or zoom

- Choose Tools > Pan & Zoom Tool or press z, then:
  - To interactively pan, click and drag in the dopesheet.
  - To zoom in, hold down the middle mouse button.
  - To zoom out, hold down the right mouse button.

⚠️ If you press Shift while this tool is active, you can switch to the rectangular zoom tool.

To interactively pan or zoom

- Choose Tools > Navigate Tool or press s, then:
  - To interactively pan, left-click and drag in the dopesheet.
- To interactively zoom horizontally, middle-click and drag to the left to zoom in, and drag to the right to zoom out.
- To interactively zoom vertically, right-click and drag up to zoom in, and down to zoom out.

If you press Ctrl while this tool is active, you can switch to the rectangular zoom tool.

To pan or zoom vertically

- Choose Tools > Vertical Pan & Zoom Tool, then:
  - To pan vertically, click and drag in the dopesheet.
  - To zoom in vertically, hold down the middle mouse button.
  - To zoom out vertically, hold down the right mouse button.
  - To zoom interactively, press Shift and hold down any mouse button. Moving up/right zooms out, and moving down/left zooms in.

To pan or zoom horizontally

- Choose Tools > Horizontal Pan & Zoom Tool or press x, then:
  - To pan horizontally, click and drag in the dopesheet.
  - To zoom in horizontally, hold down the middle mouse button.
  - To zoom out horizontally, hold down the right mouse button.
  - To zoom interactively, press Shift and hold down any mouse button. Up/right movements zoom in, and down/left movements zoom out.

Framing

Framing sets the zoom and pan so that the tracks fit in the dopesheet within a specified range.

To frame selected regions and tracks

- Choose View > Frame Selection or press f. If only regions are selected, only they are framed in the dopesheet.

To frame all keys in the dopesheet

- Choose View > Frame All or press a.

To frame all tracks within the timeline

- Choose View > Frame to Timeline or press r. This is the same as resetting the view.
Working with Tracks

The dopesheet displays a series of animation tracks—one for each parameter—for the selected objects. A track consists of a series of rectangular boxes, each representing one frame. The name of the animated parameter is displayed above its track.

When you open the dopesheet, the tracks for each animated parameter of the selected object are collapsed. If you add any new objects, the tracks for those new objects are also collapsed.

When a track is collapsed, it shows all of the keys it finds for all the parameters that are collapsed, as if the tracks were transparent and drawn on top of each other. Any overlapping keys are shown in purple and any manipulation you do on this track is actually done on each of the “underlying” tracks.

The Summary Track

At the top of the dopesheet, there is a summary track for all objects displayed in the dopesheet. This track makes it easy to branch-select a whole character to display the keys on all animated parts. Or branch-select the scene root or a model node to display all keys on all objects in the scene or model to retime the entire animation.

The summary track is a special type of collapsed track. As with other collapsed tracks, purple keys represent overlapping keys, and RGB keys represent the XYZ keys for the transformation parameters. Unlike other collapsed tracks, however, you cannot expand the summary track.

You can display the frame numbers for all or just first/last keyframes on this track by choosing Edit > Display all frames for summary track. You can set a preference for which frame numbers are displayed (see Setting Preferences for the Dopesheet on page 243).
**Expanding Tracks**

When you open the dopesheet, the tracks for the selected object are collapsed by default to save vertical space. However, you can expand them to see the track for each animated parameter of the selected object.

If no tracks are selected and you choose an **Expand** command, all tracks are expanded by groups.

*To expand tracks*

- Select one or more tracks and use one of these methods:
  - Press **E**.
  - Click the **Expand** button on the command bar.
  - Choose **Edit > Expand** from the command bar.
  - Right-click on a collapsed track and choose **Expand**.
  - Click on the collapsed track name to expand each group.

For example, clicking on an object’s kine.local track expands to the three transformation tracks (pos, rot, and scl), and clicking on one of these tracks (such as kine.local.pos) expands to show its X, Y, and Z parameter tracks (shown with red, green, and blue keys, as below).

![Tracks expanded for selected parameters.](image)

**Collapsing Tracks**

You can collapse the tracks at different levels so that you see different groupings. For example, you could collapse all the Rotation tracks into one track or collapse all the Local Transform tracks into one track.

If no tracks are selected and you choose a **Collapse** command, all tracks are collapsed into groups.

*To collapse tracks*

- Select one or more tracks and use one of these methods to collapse them:
  - Press **C**.
Working with Tracks

- Click the Collapse button on the command bar.
- Choose Edit > Collapse from the command bar.
- Middle-click on an expanded track name (branch collapsing). For example, if you middle-click on any of the Rotation X, Y, and Z tracks, they collapse into one Rotation track.
- Choose Edit > Collapse Objects Individually to group and collapse the tracks according to their path name (separates pos tracks from rot tracks, for example).
- Select two or more selected tracks, right-click on one of them, and choose Collapse or Collapse Selected Tracks.
- Choose Edit > Collapse All Tracks to collapse all tracks into one.

Displaying or Hiding Tracks

To display or hide tracks
Use one of these methods:

- Select a parameter in the animation tree to display its tracks. If a track is hidden, selecting its parameter unhides the track.
- Choose View > Hide Track, press h, or right-click on a track and choose Hide to hide the selected track.
- Choose View > Unhide All Tracks, press Shift+h or right-click the dopesheet background and choose this command to display all hidden tracks.
- Choose View > Clear All Tracks to clear all tracks from the dopesheet (this does not delete the animation). This effectively hides all the tracks and switches to View > Selected Parameters. Click on the parameters in the property tree (on the left) to view the tracks again.

Aligning the Tracks with the Parameters in the Tree

To make it easier to see which parameter is associated with which track, you can have the tracks be automatically aligned with the parameters displayed in the tree by choosing Edit > Align Tracks with Tree View. When you scroll in either the tree or track view, the other one scrolls accordingly so that the parameters and their tracks remain aligned.

In the dopesheet preferences property editor, you can set a default to have this command active or not (see Setting Preferences for the Dopesheet on page 243).
Chapter 8 • Editing Animation in the Dopesheet

The labels above each track are not displayed, but the track is displayed exactly across from the parameter name.

The local track is a summary track of the parameters beneath it, and the kine track is a summary track of all the parameters beneath it (this could be a mixture of local and global kinematics parameters).

![Diagram of tracks and parameters]

**Marking Parameters Using Their Tracks**

In addition to the other ways of marking parameters for animation, you can also use the dopesheet to do this.

*To mark parameters in the dopesheet*

- Alt+click the track's label. The label and its corresponding parameters in the tree turn yellow.

![Diagram of marking parameters]

When a track is collapsed, clicking its label marks all the corresponding parameters of all tracks within the collapsed track. When only some of the parameters in it are marked, the label (not just the marked parameter names) is marked in a lighter yellow.
Working with Keys

Unlike the fcurve view (graph) in the animation editor, where time is displayed as a continuous curve, the dopesheet shows time in discrete chunks (frame by frame). Even though the keys do not really have a duration, they are displayed in the dopesheet as if they do.

Keys are represented in the dopesheet as colored frames on the tracks. Keys can be in these colors:

- Red, green, and blue for XYZ transformation parameters and RGB color parameters.
- Black for keys other than XYZ or RGB.
- Purple for overlapping keys when tracks are collapsed.

To move a key

- Make sure the Selection tool is active (press the space bar), then click and drag the key. A yellow box appears around its frame with the frame number above it.

While you can move only one key at a time, you can move a key on a collapsed track to move all the keys on that frame for all tracks that are contained within the collapsed track.

To copy a key

- With the Selection tool active, Ctrl+click and drag the key to copy it to another frame. Release the mouse button to paste it at that frame.

To edit multiple keys

- You must create regions that contain them (see Editing Keys Using Regions on page 252).
In the dopesheet, you can edit your animation by defining regions of keys and by moving (see Moving Regions on page 254), scaling (see Scaling Regions on page 255), and copying, cutting, and pasting these regions (see Cutting, Copying, and Pasting Regions on page 256). To edit multiple keys, you must draw regions that contain them.

The region is displayed as a white rectangle on the track with a position handle (in the center) and four scale handles (left, right, top, and bottom). The region also shows the start and end frame numbers.

To draw regions
Choose Tools > Region Tool or press q, then do one of the following:

- Drag the mouse pointer in either direction over the frames and tracks you want to edit.
- To draw multiple regions (on the same or different tracks), press Shift and draw the regions, one at a time.
- To draw a region that spans multiple tracks (collapsed or not), drag over the whole area over any number of tracks.

These regions can also overlap other regions: the region creation order is the order in which they are processed. If you have keys in multiple regions, they could be offset one or more times depending on how many regions they are in.

To resize a region
- Drag any of the region's scale handles.
- If you have multiple regions selected:
  - Pressing Alt while dragging a handle resizes only that region.
- Resizing vertically (dragging the top or bottom handle) resizes only the region that you are manipulating.

To deselect regions

- Do one of the following:
  - Draw another region or click anywhere in the dopesheet.
  - Right-click on a region and choose Deselect Region.
  - Choose the Edit > Deselect All Regions command to deselect all regions (without affecting the keys).
  - Click the Deselect button on the command bar to deselect all regions and tracks.

Rippling

Use the Ripple icon on the command bar to control whether keys on the track are pushed along the track or not when you move, scale, cut, or paste regions.

When Ripple is on and you perform any of these operations, the keys are pushed accordingly, either to the left or the right. For example, if a region of 10 frames is scaled to 15 frames to the right, the keys on the right are pushed in that direction by 5 frames. Any offset between the region and the other keys is preserved.

To activate Ripple

- Click the Ripple icon (turns light blue) before you move, scale, cut, or paste a region. When Ripple is on, keys automatically snap to frame boundaries.

You can set a default for the Ripple mode state in the dopesheet preferences property editor (see Setting Preferences for the Dopesheet on page 243).

Deleting Keys from Regions

Deleting keys removes the animation from that frames that are contained in the region. Deleting does not put the keys in the clipboard buffer, but you can undo this action like anything else in XSI by pressing Ctrl+z.

To delete keys from regions

1. Select one or more regions.
2. Do one of the following:
   - Choose Edit > Delete Keys.
   - OR
- Click the Del button on the command bar.

  or

- Press Delete.

  or

- Right-click on a region and choose Delete Keys.

### Moving Regions

When you move a region, you are changing the location of the animation. You can move the region with or without affecting the keys on that track, depending on whether Ripple is on or off.

You can also move a region that has no keys in it (an empty region) and have it affect the keys on the track (with Ripple on). This is useful for easily shifting keys in long animations.

**To move regions**

- To move a region but not the keys, including to another track, left-click the position handle (+) in the middle of the region and drag. If you have multiple regions selected, you can only move them together horizontally on their respective tracks.

- To move a region without changing keys outside the region, middle-click the region’s position handle and drag with Ripple turned off.

- To move the region and push other keys (ripple) to the right (forward in time) or left (back in time), middle-click the region’s position handle and drag with Ripple turned on.

- To move multiple regions, drag one of the regions and all other regions also move with it.

- To move only one region when multiple regions are selected, press Alt while dragging the region.
Scaling Regions

When you scale a region, you speed up or slow down the animation within it. Increasing the region's size slows down the animation by increasing the number of frames between keys, while reducing the region's size speeds up the animation by decreasing the number of frames between keys.

When you scale a region, the keys within it adjust proportionally to accommodate the new size. The pivot for the scale is the opposite corner of the region.

You can also scale a region that has no keys in it (an empty region) and have it affect the keys on the track (with Ripple on). This is useful for easily shifting keys in long animations.

- You can also do mass scaling of animation using the Sequence commands in the Animation menu on the Animation panel. See Scaling and Offsetting (Retiming) Animation on page 55 for more information.
- If you want to scale all the animation in a scene using the dopesheet, branch-select the scene root and click the Collapse button (with no tracks selected). This creates one track for the whole scene which you can then resize using a region.

To scale regions

- To scale a region without changing keys outside the region, middle-click+drag its left or right handle with Ripple turned off.

  With Ripple off, the region cannot be stretched any farther than neighboring keys.

- To scale the region and push other keys (ripple) to the right (forward in time) or left (back in time), middle-click+drag either of the region’s handles with Ripple turned on. If you shrink the region (scale down), the keys outside it are “pulled” together.

- To scale multiple regions, middle-click+drag one of the region’s handles and all other regions are also scaled.

- To scale only one region when multiple regions are selected, press Alt while dragging the region’s handles.
Cutting, Copying, and Pasting Regions

You can copy, cut, and paste regions of frames (with or without keys) on a track to edit animation. Cutting removes the frames from the regions and puts them in the paste buffer, while copying a region makes a copy of it and puts it into a paste buffer. Because the region you have cut or copied remains in the paste buffer until it is replaced, you can paste the same region as many times as you like.

You can copy key values and interpolation information between different parameters for the same object or for different objects. For example, you could copy the X rotation values from one object to the Y scaling of another object.

Once the region is in the paste buffer, you can reinsert it in another area of a track. You can click anywhere on a track to create an insertion point at which the region is pasted. You can also paste the region within a defined region, in which case the pasted animation is scaled up or down to fit into the new region and replaces its original contents (where the keys overlap).

Copying and pasting regions without keys (empty regions) with Ripple on is an easy way to offset the animation. The number of frames in the blank space you copy doesn’t matter: it’s the size of the region into which you paste the blank frames that determines the amount it ripples.

You can display and select any unanimated parameter in the animation tree (see Controlling What Is Displayed in the Dopesheet on page 244) to have an empty track displayed for it. This makes it easy to copy regions of keys from an animated parameter and paste them on the track of an unanimated parameter.

Cutting or Copying a Region

To copy or cut a selected region

1. Make sure that Ripple is on if you want to push other keys on the track, or turn Ripple off if you want to paste over other keys on the track.

   If you cut a region with Ripple on, the keys that come after the region will move to the left (back in time) to fill in the void.

2. Select one or more regions of frames (with or without keys) you want to copy or cut.

3. Cut or copy the regions to the clipboard using any of the following methods:

   - Ctrl+middle-click and drag to copy the region of keys.

     Keep the mouse button pressed down as you drag the region to another location, then release the mouse button to paste the region there.

     or

   - Press Ctrl+c (Copy) or Ctrl+x (Cut).
Cutting, Copying, and Pasting Regions

- Click the Copy or Cut button on the command bar.

- Right-click on the region and choose Copy Keys or Cut Keys.

- Choose Edit > Copy Keys or Cut Keys from the command bar.

Pasting the Region

There are different factors that determine the results of pasting keys.

- If Ripple is on and you click to paste the keys at the point of insertion, the frames to the right of the inserted region move further to the right (ripple forward in time) to accommodate that region.

- If Ripple is off, the keys are pasted in the region or after the point of insertion (existing keys are either overwritten or merged).

If you are pasting in a region, you can choose to either merge or replace the keys in the destination region—see Merging or Replacing Keys When Pasting on page 258.

To paste the region

1. Do either of the following:

   - Click at the desired insertion point in a track.

   To create an insertion point across multiple tracks for pasting, middle-click+drag vertically across the tracks you want. The region is locked in X direction (horizontally).

   

<table>
<thead>
<tr>
<th>Click on the track where you want to insert the pasted keys.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Middle-click+drag to create an insertion point across multiple tracks</td>
</tr>
</tbody>
</table>

   - Create a region into which you will paste the cut or copied keys.
When you paste the keys, they will be scaled up or down to fit into that region.

2. Paste the region from the clipboard using any of these methods:
   - Choose Edit > Paste Keys from the command bar.
     or
   - Click the Paste button on the command bar.
     or
   - Right-click on the region and choose Paste Keys.
     or
   - Press Ctrl+v.

   The region is copied or cut from the paste buffer and reinserted into a track. The insertion point is at the end of the pasted region, making it easy to paste one region after the other.

3. You can continue pasting the clipboard contents to other regions.

**Merging or Replacing Keys When Pasting**

When you paste keys with the Ripple off, you can choose to either merge the cut/copied keys with the existing keys in the destination region, or you can replace the existing keys with the cut/copied keys.

**To merge or replace keys when pasting**

1. Select the region into which you want to paste the keys.

2. Choose either of these commands:
   - Edit > Merge Keys When Pasting. Cut/copied keys are merged with any existing keys in the destination region.
Cutting, Copying, and Pasting Animation between Tracks

Although you can copy/cut and paste regions from a lower-level track to a higher-level track, such as from the animation-parameters level to the animation-track level (for example, from sclx to Scale), there are limitations. This applies to tracks on the same or another object.

You can copy/cut and paste regions from one object’s track to another’s only if the track level is the same (such as Scaling, Rotation, Position) or higher and the animation attribute (such as XYZ) or relation are the same.

For example, you can copy/cut an object’s Rotation tracks (X, Y, and Z) and paste it to another object’s Rotation tracks (X, Y, and Z), but not to its Scaling tracks. Or you could copy an object’s Rotation Y track and paste it to another object’s Rotation track.

An exception to this occurs at the lowest-level tracks (attribution parameters and target relations level). At these levels, you can copy from one attribute parameter type to another; for example, you can copy one object’s Rotation Y track to another object’s Scaling X track.

You can also copy and paste between local and global transformation tracks. For example, you could copy a chain’ effector’s keys on its Global - Position tracks (X, Y, and Z) and paste them on a null’s Local - Position tracks (X, Y, and Z).

Cutting, Copying, and Pasting Animation between Models and Hierarchies

You can cut/copy and paste animation from one model to another or from one hierarchy of objects to another within the same model. The objects in the hierarchy or model must have the same parameter names (or a subset or superset of them).

This is similar to copying actions from one model to another (see Copying Action Sources between Models on page 172 in the Nonlinear Animation guide). For example, you can paste a walk cycle animation from the Bob model to the Fred model as long as Fred has the same parameter names as Bob.

If you have submodels, their animation won’t be copied and pasted.
To copy animation between models

1. Select a model node in the explorer.
2. In the dopesheet, create a region on the summary track containing all the keys.
3. Cut or copy the region.
4. Select the model to which you want to copy the animation and update the dopesheet.
5. Create a region in the summary track for this model and paste the keys.
Deactivating (Muting) Animation in the Dopesheet

You can use the dopesheet to temporarily deactivate animation or an operator so that it has no effect on the object. Deactivating the animation or operator doesn’t remove it from the object.

You can deactivate tracks and regions for any type of animation, including function curves, constraints, expressions, and deform operators (including envelopes).

Constraints and expressions have an Active parameter while deform operators have Mute parameters (deformations) that can be on or off (they are considered Boolean). See Deactivating Boolean Parameters on page 263 for more information on these parameters.

Deactivating Tracks

You can activate and deactivate a parameter’s animation by clicking the activation box at the left of its track or by choosing a menu command.

The activation box is green when a parameter’s animation is active and red when the animation is inactive. If the tracks are collapsed and there is any combination of active and inactive tracks, the box is stippled green and black.

The activation box for an unanimated parameter’s track is gray to indicate that there is no animation to be activated/deactivated.

To deactivate a track

- Do one of the following:
  - Click the green box at the left end of any track so that it turns red.

or
Chapter 8 • Editing Animation in the Dopesheet

- Select one or more tracks and choose Edit > Deactivate Tracks.

To activate a track

• Do one of the following:
  - Click the red box at the left end of any track so that it turns green.
    or
  - Select one or more tracks and choose Edit > Activate Tracks.

Deactivating Animation in Regions

You can deactivate a region of any track so that the frames in that region don’t contribute to the animation. Deactivated regions are displayed in dark gray.

Deactivated regions appear in dark gray.

To deactivate animation in a region

1. Activate the Region tool (press q) and draw one or more regions.
2. Do one of the following:
   - Click the Deactiv button on the command bar.
     or
   - Right-click in the region and choose Edit > Deactivate Region.
     or
   - Choose Edit > Deactivate Region from the command bar.

To activate animation in a region

1. With the Region tool active, draw one or more regions over deactivated frames.
2. Do one of the following:
   - Click the Activ button on the command bar.
     or
   - Right-click in the region and choose Edit > Activate Region.
     or
   - Choose Edit > Activate Region from the command bar.
Deactivating Boolean Parameters

Boolean parameters are parameters that can be either on or off. Constraints and expressions have an Active parameter that is Boolean, while deform operators (deformations) have a Mute parameter that is Boolean. Because the Active and Mute parameters have the opposite meaning, you must be careful to what the state of the tracks and regions mean in each case (see the parameter descriptions in this section for information).

Boolean tracks have different colors and appearance from regular fcurve tracks. When they are inactive, their frames are dark gray and shown on the bottom half of the track; and when they are active, their frames are a light sand color and shown on the upper half of the track.

For an unanimated Boolean parameter, clicking the red/green activation box at the end of its track toggles the constraint or deformer’s activeness. For example, if you display a track for the position constraint’s unanimated Active parameter and you click its activation box, you toggle the constraint’s activeness.

To see the Active or Mute parameters in the dopesheet
Make sure that you have the correct filter selected in the animation tree’s View menu (the View menu on the furthest left):

- To see the Active parameter for constraints, choose View > Animation Properties.
- To see the Mute parameter for Deform operators, choose View > Modeling Properties.

Active Parameters
When an Active parameter is inactive, the constraint or expression operator is also inactive; when an Active parameter is active, so is the operator.

To activate/deactivate Active parameter tracks
- Deactivate and activate Active parameter tracks as you do for any parameter (see Deactivating Tracks on page 261).

When the Active parameter’s track is deactivated, so is the operator.

To activate/deactivate Active parameters in a region
- Deactivate and activate regions for the Active parameter as you do for any parameter (see Deactivating Animation in Regions on page 262).

When the Active parameter’s region of frames is deactivated, the operator is also deactivated over those frames.
Chapter 8 • Editing Animation in the DopeSheet

When a Mute parameter for a deform operator is inactive, it means that the deform operator is active; when Mute is active, the operator isn’t.

To mute/unmute Mute parameter tracks

- Deactivate or activate tracks as you do for any parameter (see Deactivating Tracks on page 261).

However, when the Mute parameter track is deactivated, the deform operator is then active. Likewise, when the Mute track is active, the deform operator is not active.

To mute/unmute Mute parameters in a region

- Create a region of frames and do either of the following:
  - Click the Mute button on the command bar to mute (deactivate) the deform operator.
  - Click the Unmute button to unmute (activate) the deform operator.

Mute parameter for a Bulge operator with a region that is active (sand colored). This means that the operator is muted.

Mute parameter with a region that is inactive (dark gray), which activates (unmutes) the Bulge operator over these frames.

Click box to toggle track’s activation state.
Chapter 9  

**Animating along Paths and Trajectories**

A path provides a route in global space for an object to follow in order to get from one point to another. The object stays on the path because its center is constrained to the curve for the duration of the animation.

In XSI, path animation is actually a type of constraint, unlike in SOFTIMAGE|3D. For more information on constraints in general, see *Chapter 10: Animating with Constraints* on page 283.

This plane uses **path** animation. Its position is measured as a percentage along the curve.

This plane uses **trajectory** animation. It jumps from knot to knot at each frame.
You can create path animation using a number of methods, each one having its own advantages. You can:

- Set an existing curve as the path for an object—the object is automatically animated along the percentage of the curve’s length. See Setting Objects on Paths on page 268.

- Constrain an object to curve using a predefined constraint and set keys for the percentage of the path traveled. See Constraining an Object to a Curve (Path) on page 272.

- Pick a trajectory to use a curve’s knots as indicators of the object’s position at each frame. See Setting Objects on Trajectories on page 274.

- Move an object about your scene and save path keyframes at different positions—the path curve is created automatically as you go. See Creating a Path from Path Keys on page 277.

- Convert the existing movement of an object into a path. See Converting an Object’s Translation to a Path on page 278.

After you’ve created path animation, you can modify the animation by changing the path curve itself or by changing the timing of the object on the path. See Modifying the Path Curve or Trajectory on page 281 and Modifying the Path Timing on page 279, respectively.

To store the path animation in an action to use in the animation mixer, see Storing Path Animation on page 104 in the Nonlinear Animation guide for more information.

**Viewing Path and Trajectory Information**

XSI provides a visual reference to path and trajectory information such as key points, types of constraints, and constraint options. As well, there are certain nodes in the explorer that are particular to paths.

To create and view a curve from the trajectory or path of an animated object, plot its animation as a curve—see Creating a Curve by Plotting an Object’s Transformation on page 274 for more information.

You can also use ghosting to view an object’s path as a motion trail—see Ghosting Animated Objects on page 42.

To display path (constraint) information in a viewport, click the eye icon and choose Relations from the viewport title bar. A dashed line appears, connecting the center of the selected object and its constraining path. Key points and their corresponding frame numbers are also displayed:

- A square represents a key saved on the path.

- A triangle represents a locked path key.

- A circle represents a key set directly from a property page or the animation editor. These are the only type of keys found on trajectories.
If there is an up-vector, it is indicated by an arrow. The dotted line is connected to the center of the constraining curve. You can select the line and press Enter to open the PathCns or TrajectoryCns property editor.

A triangle represents a locked-path key.

A square represents a key saved on the path.

A circle represents a key set directly from a property page or the animation editor. These are the only type of keys found on trajectories.

For finer control of this display, open the Visibility Settings property editor by clicking the eye icon and choosing Visibility Options, then click the Attributes tab.

You can then display constraint information independently for selected and unselected objects using the Relations (links and key points) and Relations Info (frame numbers and other labels) options.

**Finding Path Elements in the Explorer**

In the explorer, expand the path-animated object’s Kinematics > Constraints branch to see the PathCns or TrajectoryCns node. Path animation is stored here because it is actually a constraint.

You can click the node’s icon to open the PathCns or TrajectoryCns property editor for changing the path timing, as well as any other attributes.
Setting Objects on Paths

The quickest and easiest way of animating an object along a path is by using the Set Path command and picking the curve to be used as the path. When you use this command, the animation is created for you. There’s no need to set keyframes—just set the start and end frames.

A percentage (perc) function curve is created with your choice of spline (ease-in/ease-out) or linear interpolation. After you have created path animation, you can modify the timing by editing this function curve (see Modifying the Path Timing on page 279) or you can change the path curve itself to change the animation (see Modifying the Path Curve or Trajectory on page 281).

Want to convert a path animation to translation? Plot the position of the path-animated object, then apply the result to the object or as an action in the animation mixer. See Plotting Animation on page 59 for more information.

Picking a Path

To animate an object along a path

1. Create a curve using any of the available tools in the Curve menu on the Model or Animate toolbars.

   The parameterization of the curve has no effect on timing.

2. Select the object you want to animate.

   Make sure the selected object does not have compensation on when creating a path: this would key the offsets rather than path animation. For more information on compensation, see Creating Offsets between Constrained and Constraining Objects on page 318.

3. Choose Create > Path > Set Path from the Animate toolbar.

4. In the dialog box that opens, enter the frame at which the object starts moving (Start Frame) and the frame at which the object reaches the end of the path (End Frame). By default, the object travels the full length of the path.

   - If you want the object to move along the path at a constant speed, select Linear. Otherwise, the object will ease into and out of the animation.
- If you want the object’s X axis (by default) to stay aligned with the path’s slope, select **Tangent**. The constrained object’s axis is repositioned to point in the direction of the curve’s slope. As the object moves along the path, its X axis follows the curve’s slope. For more information on tangency constraints, see **Tangency Constraints** on page 316.

5. Click OK and pick the curve to use as the path. Keys are created at each end of the path, and the object’s intermediate position along the path is interpolated.

6. After you pick the path curve, the PathCns property editor opens, allowing you to manually adjust the settings.

   The **Path %age** (percentage) slider is especially important for controlling how much of the curve is used as a constraining force on the selected object.

7. If you selected the **Tangent** option in the Apply Path dialog box, the **Active** option is selected on the Tangency page.

   You can also set the X, Y, and Z values to define a vector along which the object’s center is aligned (the default is X). Set the appropriate slider to 1 to keep that axis tangent to the path.
Chapter 9 • Animating along Paths and Trajectories

To modify the path timing, see Modifying the Path Timing on page 279; to modify the path curve itself, see Modifying the Path Curve or Trajectory on page 281.

To remove the animation, see Removing Path or Trajectory Animation on page 282.

Example: Path through a City

This example shows you how to apply path animation to the camera and its interest.

Create a cityscape using cubes and a grid

1. Choose Get > Primitive > Polygon Mesh > Grid and Get > Primitive > Polygon Mesh > Cube to create objects for a cityscape.

2. Using Edit > Duplicate Multiple, create several copies of the cube. Scale them so that they are all different sizes and place them on the grid, using the illustration on the left as a reference.

Create a path for the camera and its interest

3. Choose Create > Curve > Sketch from the Model or Animate toolbar and draw a freeform curve path that goes through the city.

Animate the camera along the path

4. Change the timeline’s End Frame to 300.

5. To select the camera, do one of the following:
   - Open the explorer and select the camera.
   - Click the camera in a viewport. If the camera is hidden in a viewport, click the eye icon in the viewport title bar and choose Cameras, and then select the camera from the list.

6. Choose Create > Path > Set Path from the Animate toolbar. Use the default values for the start and end frames, select Linear, and click OK.

7. Pick the curve as the path.
8. Play back the animation. You will notice that the camera is following the path but the camera interest stays fixed.

**Animate the camera’s point of interest on the path**

9. Select the camera interest by clicking it in a viewport or the explorer.

10. Choose Create > Path > Set Path, and set the start and end frame values to –20 and 280. Pick the curve path.

11. Play back the animation. The camera interest is now traveling along the path slightly ahead of the camera.
Chapter 9 • Animating along Paths and Trajectories

Constraining an Object to a Curve (Path)

You can constrain an object to a curve and animate its position on that curve. This constraint is particularly useful for having an object follow a path’s direction, such as a roller coaster car following the tracks.

The Constrain > Curve (Path) command applies the same PathCns constraint as the Set Path command described in Setting Objects on Paths on page 268, but you must manually keyframe the percentage of the path traveled—it’s not calculated automatically for you.

To constrain an object to a curve as a path

1. Create a curve using any of the available tools in the Curve menu on the Model or Animate toolbars.
2. Select the object you want to constrain to the curve.
3. Choose Constrain > Curve (Path) from the main command panel.
4. Pick the curve to be the path. By default, the constrained object’s X axis is repositioned in the direction of the slope of the curve. As the object moves along the path, its X axis follows the curve’s slope.
5. In the PathCns property editor that opens, use the Path Percentage slider to animate the object along the path.

Move the playback cursor to the appropriate frame on the timeline, set a percentage value on the Path Percentage slider, and click the animation icon (green box) beside the slider.

Continue like this to add as many keyframes as you wish.

- For information on the other options in this property editor, see Setting Objects on Paths on page 268.
- To modify the path timing, see Modifying the Path Timing on page 279; to modify the path curve itself, see Modifying the Path Curve or Trajectory on page 281.
- To remove the animation, see Removing Path or Trajectory Animation on page 282.
Chapter 9 • Animating along Paths and Trajectories

Setting Objects on Trajectories

Trajectories are similar to paths in that they are specialized constraints that control an animated object’s translation. Unlike a path, which interpolates an object’s position between keyframes, a trajectory’s knots serve as indicators of the object’s position at each frame. For example, if a trajectory has 18 knots, it takes only 18 frames for the object to complete its journey on the path.

A trajectory requires no keys because the object jumps from one knot to the next on the curve at each frame. This means you have absolute control as to where your object will be from frame to frame.

A situation in which you’d want to use trajectories instead of a path is when combining 3D animation with live action. You need to control the exact position of an object at each frame, which you can do with a trajectory.

You can create trajectory animation by picking an existing curve as the trajectory for an object and selecting the starting frame for the animation.

Creating a Curve by Plotting an Object’s Transformation

To quickly create a curve with the necessary knots for a trajectory, you can animate an object’s position and plot the motion of the object’s center to generate a curve. Plotting a curve helps you visualize the trajectory of an animation. You can then edit the curve to fine-tune the trajectory position for each frame.

You can plot the movement of a selected point or cluster, similar to the Tag2Path command in SOFTIMAGE|3D.

You can also use ghosting to view an object’s trajectory as a motion trail that is updated as the animation changes—see Ghosting Animated Objects on page 42.
Setting Objects on Trajectories

To create a curve from an animation
1. Select an object, point, or cluster whose transformation parameters are animated.
2. Choose Tools > Plot > Curve from the Animate toolbar.
3. In the Plot Curve dialog box that opens, set these options:
   - The Start Frame and End Frame specify the frames at which to begin and end plotting.
   - The Step Value controls the sampling of the generated curve. If this value is 1, a point on the curve is created for every frame; if this value is 2, a point is created for every other frame; and so on.
   - Curve Name is the name to use for the curve that is generated from plotting the transformation.
4. When you click OK, the animation is played back and a curve is created showing the path or trajectory of the animated object. The original animation remains.

Setting an Object on a Trajectory

You can create trajectory animation by picking an existing curve as the trajectory for an object and selecting the starting frame for the animation.

To set a trajectory
1. Create a curve using any of the available tools in the Curve menu on the Model or Animate toolbars. The number of points (knots) on the curve determines how long it takes the object to complete its course on the path.
2. Select the object you want to animate.
3. Choose Create > Path > Set Trajectory from the Animate toolbar.
4. Pick the curve.
5. In the Trajectory Constraint property editor that opens, set the animation’s Start Frame.
   For information on the other options in this property editor, see Setting Objects on Paths on page 268.
6. Play the animation to check the results and timing.

To remove the animation, see Removing Path or Trajectory Animation on page 282.
Modifying the Animation

You can either modify any values in the TrajectoryCns property editor or directly modify the trajectory curve (see Modifying the Path Curve or Trajectory on page 281).

To modify the animation

1. Select the object on the trajectory.
2. Open the TrajectoryCns property editor.
3. Change the Start Frame to change the timing or change any other value that you initially set.
Creating a Path from Path Keys

You can define path animation by moving an object and saving path keys at different frames. The path curve is created automatically as you go. The keys can either represent a percentage of the path length or be locked to a particular point along the path, depending on which command you choose.

Looking for SOFTIMAGE|3D path equivalents in XSI? Look no further!

- Path > Save Key on Path is the same as
  SaveKey > Object > Translation in SOFTIMAGE|3D.

- Path > Save Locked Key on Path is the same as
  SaveKey > Object > Key Path Translation in SOFTIMAGE|3D.

To create a path by setting path keys

1. Move the playback cursor to the frame where you want the animation to begin and select the object you want to animate and translate it to where you want it to start.

2. Choose one of the following commands from the Path menu on the Animate toolbar:
   - Save Key on Path sets a key corresponding to a percentage of the path's length.

   If you modify the path later by adding, deleting, or moving its points, the timing may change because the same percentage would correspond to a different location on the curve.

   or

   - Save Locked Key on Path sets a key corresponding to a specific location on the curve. The object always passes through that specific location on the curve at that frame, even if the shape or length of the path changes.

3. Change frames, move the object, and repeat step 2 (middle-click the Path menu button to repeat the last-used command).

   XSI automatically updates the curve and timing information. Continue until you have completely defined the path.

   - To modify the path timing, see Modifying the Path Timing on page 279; to modify the path curve itself, see Modifying the Path Curve or Trajectory on page 281.
   - To remove the animation, see Removing Path or Trajectory Animation on page 282.
Chapter 9 • Animating along Paths and Trajectories

Converting an Object’s Translation to a Path

You can use the Create > Path > Convert Position Fcurves to Path command to convert part or all of an object’s existing translation animation into path animation.

Want to do the reverse and convert path animation to translation? Plot the position of the path-animated object, then apply the result to the object or as an action in the animation mixer. See Plotting Animation on page 59 for more information.

To convert an object’s translation to a path

1. Select the translated object.
2. Choose Create > Path > Convert Position Fcurves to Path from the Animate toolbar.
3. In the Convert Position Animation to Path dialog box, enter the Start Frame and End Frame to set which segment of the function curve will be used to create the path. Any animation before the Start Frame or after the End frame will be discarded.
4. Enter the desired interval between keyframes as the Step.

XSI creates the path with knots based on the object’s location at the Step interval. For example, if the start and end frames are 10 and 20 and the Step is 5, the curve will be created with knots based on the object’s location at frames 10, 15, and 20.

If you want a keyframe on every frame, use a Step value of 1.

Because the path is a curve, an extra knot will be created after the first one and before the last one.

XSI replays the chosen segment of the animation and creates the new path.

If you have a large number of keys on the resulting curve, you can use the Modify > Curve > Clean command on the Model toolbar to reduce the number of points while keeping the curve’s general shape.

You can also use the Create > Curve > Fit on Curve command which creates a second curve instead of modifying the first one.

See the Modeling and Deformation Basics guide for more information on these commands.

• To modify the path timing, see Modifying the Path Timing on page 279; to modify the path curve itself, see Modifying the Path Curve or Trajectory on page 281.

• To remove the animation, see Removing Path or Trajectory Animation on page 282.
Modifying the Path Timing

After you’ve created path animation, there are three ways in which you can modify the timing:

- Use the Path Retime command.
- Set keys for the Path percentage in PathCns or TrajectoryCns property editor.
- Edit the Percentage fcurve in the animation editor.

Retiming the Path Animation

To retime the path animation

1. Select one or more path-animated objects.
2. Choose Path > Path Retime from the Animate toolbar.
3. In the Sequence dialog box, make sure the Retime option is selected.
4. Set the frames of the path animation that you want to use with the Initial Range Start and End frames.
5. Specify the Final Range Start and End frames that you want to use for the new timing.

For example, you can shorten (and therefore increase the speed) a path animation that went from frame 1 to 100 to frames 20 to 70. You can even reverse the animation—for example, enter 100 as the start and 1 as the end frame.

![Affected/Initial Range (frames)]

The path percentage is updated with the new values.

Setting Keys or Editing the Path Percentage

To edit the Path Percentage parameter or function curve

To modify the path timing, you can directly adjust the Path Percentage in the PathCns property editor and set keys at different frames or edit the Percentage function curve in the animation editor.
1. Select the object on the path.

2. Adjust the path percentage by doing one of the following:
   - Display the PathCns property editor and adjust the Path %age values.
   - Open the animation editor with the Percentage function curve already selected. To do this, right-click the Path Percentage's animation icon and choose Animation Editor.

   ![PathCns property editor]
   
   ![Animation editor with Percentage function curve]

   You can open the animation editor in any other way (in a viewport, in a floating window, etc.), but then you have to hunt for the Percentage fcurve.

   The percentage (perc) function curve is displayed with key points representing keys on the path.

   ![Percentage function curve]

   Locked keys on the path are blue—they are locked in value, but not time. You can unlock a key by selecting it and choosing Keys > Lock in Y (Value) from the animation editor command bar.

3. Set new keys for the Path Percentage parameter or edit the Percentage function curve as you would any other function curve (see Chapter 7: Editing Function Curves on page 159 for information).

   With either of these methods, you can set keys with values below 0% or higher than 100% of the path. XSI extrapolates path percentages outside of the 0 to 100% range, depending on the type of curve used as a path:

   - On a closed curve, such as a circle or a square, using path percentages less than 0% or greater than 100% will continue to translate the object on the curve, with increasing percentages resulting in clockwise movement (relative to the local front of the object).
For example, if an object were keyed to start at 0% of a circular path and end at 150%, the object would make one complete clockwise revolution of the circle and continue for another half a revolution. If the end were at –150%, the object would travel the same distance counterclockwise.

- On an open curve, such as a spiral or a NURBS curve, keys below 0% place the object at the curve’s starting point; keys above 100% place it at the curve’s endpoint.

### Modifying the Path Curve or Trajectory

To modify the path, you can edit the curve itself or translate the object constrained to the path.

For more information on editing curves in general, see *Curves* on page 25 in the Modeling and Deformation Basics guide.

**To modify a curve or trajectory**

Do one of the following:

- Move, add, or remove points on the curve as you would to edit any curve.
  
  or

- Translate the constrained object and use its new location to simultaneously create a new knot and keyframe on the path.

To do this, translate the object at the appropriate frame, then choose Create > Path > Save Key on Path or Save Locked Key on Path.

### Substituting a Path’s Curve

After you have animated an object on a path, you can substitute the original curve with a new one to create a completely different animation.

This creates the same results as the Path > Substitute command in SOFTIMAGE|3D.

1. Select the curve that is currently the path.
2. Choose Edit > Operator > Substitute Input from main command panel.
3. Pick the curve to be the new path. Right-click to end the picking session.

The object is assigned to the new curve.
Removing Path or Trajectory Animation

To remove path or trajectory animation from an object

Do one of the following:

- Select the object, then choose Constrain > Remove Constraint from the Constrain panel.
  
  If the object has more than one type of constraint, the status bar prompts you to pick the constraint to remove. If this happens, pick the constraining curve.
  
  or

- In the explorer, select the object’s PathCns or TrajectoryCns node (see Finding Path Elements in the Explorer on page 267) and press Delete.
  
  or

- If constraints are visible in a viewport (see Viewing Path and Trajectory Information on page 266), click the dotted line representing the path or trajectory constraint and press Delete.
Constraining is a way of increasing the speed and efficiency in which you animate. It's a way of animating one object “via” another one's animation. You constrain different properties, such as position or direction, of one object to another object. Then when you animate the constraining object (sometimes known as the *constrainer*), the other object’s constrained properties are animated as well.

In terms of saving time, the value of constraining objects is considerable. Suppose you want to animate a group of people at an air show watching a plane fly overhead. If you used keyframes to animate, you would have to generate function curves for each head movement as well as the plane. With constraints, however, you constrain each spectator’s head to the direction of the plane and then animate only the plane itself.
There are many ways to constrain objects, and XSI has a number of constraints packaged into predefined commands. For example, the Constrain > Direction command constrains the direction of a selected, or “aligned,” axis (the default is the X axis) of one or more objects to another object. The location of the constrained objects do not change, but their aligned axes always point at the center of the constraining object, no matter where the constraining object is.

Each type of constraint is described in detail in this chapter except for Constrain > Curve (Path), which is described in Chapter 9: Animating along Paths and Trajectories on page 265.

Global versus Local Space

One important thing to know about constraints is that constrained objects are transformed in global space. Because of this, any edits you make in a constrained object’s Local Transform property editor (such as position and rotation limits) have no effect if they are applied before a constraint: the constraint will override them.

The fact that constraints are global also matters when you want to mix constraints with other animation in the animation mixer. For more information on this, see Actions on page 89 in the Nonlinear Animation guide.

Order of Constraint Operations

In all cases, the most recently applied constraint overrides older constraints if a conflict between their properties occurs.

Expressions and scripted operators are evaluated before constraints on the same object, and the bounding constraints are applied after all the other constraints.

If you use the value of a constrained parameter in a scripted operator, you may not get the desired results. For example, suppose you want to write a scripted operator to control an object’s orientation based on its position: if the object has a path constraint, the orientation is evaluated before the position is updated at each frame, so the operator uses stale values. To achieve the desired result, you could constrain an intermediate object (such as null) to the path, then constrain the object’s position to the null. When calculating the new orientation, use the null’s position.

Multiple Constraints

You can have more than one constraint applied to a particular object at the same time. When more than one constraint is active at the same time on an object, all constraints are used but the constraint that is applied last takes priority (unless you blend the constraints—see Blending Constraints on page 321).

For example, if an object has position and orientation constraints on it, both can control the object. However, if an object has an orientation constraint and then you apply a direction constraint (which also controls the object’s orientation), the direction constraint takes priority.
About Up-Vector and Tangency Constraints

For many of the constraints, you can add a tangency or up-vector constraint to the mix. The tangency and up-vector constraints aren’t “real” constraints like the others, but are actually properties of several constraint types that determine the direction in which the constrained object should point.

Tangency constrains the X axis of an object in a specific direction, and the up-vector constrains the Y axis (up direction) of an object in a specific direction. For example, if you apply a Direction constraint to an object, you can also add an up-vector constraint to control the “roll” of the direction-constrained object.

For more information, see Up-vector Constraints on page 312 and Tangency Constraints on page 316.

Undoing a Constraint

When you remove a constraint, the object that was constrained does not return to its preconstrained state; however, if you undo the creation of a constraint (press Ctrl+z), the object does return to its pre-constrained state.
Overview of Working with Constraints

This overview shows the four basic steps needed to create any constraint in XSI.

1. Select the object to be constrained.

2. Choose the constraint command from the Constrain menu.

3. Pick the constraining (control) object.

4. Set up the constraint in its property editor.

In XSI, each constraint has its own operator with its own set of controls. For this reason, you cannot do “multi-picking” in a constraint session as you can do in SOFTIMAGE|3D. The pick session ends when you have chosen all your inputs for the constraint.

Selecting Objects with Constraints

When you have a crowded scene, it’s sometimes difficult to select only the objects you want. To make it easier to select only objects that have constraints, you can use a special constraints filter from the Filters menu on the Select panel.

To use the selection filters

1. In the Select panel of the main command panel, click the little arrow button to open the Filters menu.

2. Select the Obj w Constraint filter.
Selecting the Constrained and Constraining Objects

In addition to selecting objects that have constraints, you can also use the following commands to help you remember which objects are constrained to each other.

To select the constraining objects

- Select one or more objects and choose Constrain > Select Constraining Objs.

This command selects the objects to which the currently selected objects are constrained. For example, if A is constrained to B, selecting A and choosing this command then selects B.

This command also works when constraint properties are selected, as well as when a mixture of objects and constraint properties are selected.

To select the constrained objects

- Select one or more objects and choose Constrain > Select Constrained Objs.

This command selects the objects which are constrained to the currently selected objects. For example, if A is constrained to B, selecting B and choosing this command then selects A.
Chapter 10 • Animating with Constraints

Viewing Constraints and Their Information

You can identify the objects that have constraints on their parameters in the 3D views and in the schematic view. As well, you can use a constraint query to get information about the constraints on an object.

Viewing in a 3D View

To view constraints in a viewport

- In any viewport title bar, click the eye icon and choose Relations.
  - When you select a constrained object, a white line linking it to its constraining object is displayed. The link is labeled with the type of constraint, such as Pos for a position constraint.
  - When you select a constraining object, a light purple (lavender) line linking it to its constrained object is displayed.

For direction constraints, a link with an arrow at one end shows the relationship between the constrained and constraining object (the large arrow points to the constraining object).

To open the constraint property editor

- Select the link line and press Enter. This always selects the constraint in the context of the constrained object.

To view constraint relations information

1. Click the eye icon in a viewport and choose Visibility Options (or press Shift+s).
2. From the list of options on the Attributes page in the Visibility Settings property editor, you can choose to show Relations icons or Relations Information for selected or unselected objects, or both.

The relations information shows whether the constraint is deactivated (off), the constraint’s priority (the order in which it was applied in the case of multiple constraints—1, 2, 3, etc.), and its blended weight value (if multiple constraints are blended).

If there are multiple offsets on the object’s constraints, two arrows and their distance displays tell you the length of each link.

**Viewing in the Schematic View**

* To view constraints in the schematic view
  * Open a schematic view and choose Show > Constraint Links, then select a constrained or constraining object.

A constrained object has the letter C above its node. A green arrow with the name of the constraint type (such as Direction, Position, UpVector, etc.) links objects that are affected by a constraint.
Chapter 10 • Animating with Constraints

Animation Icon

When a parameter (global transformation) is controlled by a constraint, the parameter’s animation icon changes from its default green box to a letter C and a connection [C]. See The Animation Icon on page 26 for more information about the different states of the animation icon.

Constraint Spreadsheet Query

You can use a special constraint query in the spreadsheet to show the constraints on an object. This query makes it easy to get an overview of constraints for selected objects, as well as activating/deactivating a constraint, changing the blend weight, and seeing the constraining objects (inputs).

To select the query

1. Select one or more constrained objects.
2. Open a spreadsheet (press Alt+3) and choose Query > Constraints.

<table>
<thead>
<tr>
<th>Name</th>
<th>Constraints</th>
<th>Type</th>
<th>Active</th>
<th>Blendweight</th>
<th>UpVec</th>
<th>Active</th>
<th>Input(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>cylinder</td>
<td>kine_poss1</td>
<td>poses</td>
<td>✓</td>
<td>1.00X</td>
<td>✓</td>
<td>SecInBall</td>
<td></td>
</tr>
<tr>
<td></td>
<td>kine_poss2</td>
<td>poses</td>
<td>✓</td>
<td>1.00X</td>
<td>✓</td>
<td>dec</td>
<td></td>
</tr>
<tr>
<td></td>
<td>kine_dires</td>
<td>dieres</td>
<td>✓</td>
<td>1.00X</td>
<td>✓</td>
<td>null</td>
<td></td>
</tr>
</tbody>
</table>

For more information, see The Spreadsheet on page 106 in the Interface and Tools guide.

Finding Constraint Elements in the Explorer

You can select a constrained object and then use an explorer to find its constraint properties.

- Choose Explore > Constraints from the Select panel to open up a contextual explorer for the object.

  or

- In the explorer, press e to show only the properties of the currently selected objects.

  Expand the constrained object’s Kinematics > Constraints branch to see any constraint that is applied. Click the constraint’s name to select it, or click the constraint’s icon to open its property editor for changing any attributes, activating or deactivating it, etc.

  - Right-click the Constraints folder to open a menu where you can choose Properties to open a property editor showing all constraints in this folder. You can also activate, deactivate, or delete all constraints in that folder using the menu.
  
  - As with other operators in XSI, the constraint that is applied last is displayed at the top of the stack in the explorer.
  
  - You can’t rename constraints. However, if you have multiple constraints of the same type on an object, each constraint name is differentiated by a number in brackets, such as PosCns[3].

290 • SOFTIMAGE|XSI
Position Constraints

You can constrain one or more objects to the position (location) of another object. This can be any kind of object that has a center, including all standard geometric objects as well as nulls, lights, cameras, and lattices. The Position constraint constrains an object’s center to the constraining object’s center, moving the constrained object to that location.

Because this constraint puts the objects directly “on top” of each other, you usually need to offset the constrained object, either in the constraint’s property editor (see below) or with constraint compensation (see next page).

To constrain an object’s position

1. Select the object to be constrained.
2. Choose Constrain > Position and pick the object to act as the constraining force.

The constrained object’s center is repositioned to the center of the constraining object.

3. In the Position constraint property editor, you can set an offset between the centers of the constrained and constraining objects by entering values for the X, Y, Z sliders for either of them.

See Creating Offsets between Constrained and Constraining Objects on page 318 for more information about offsets.
4. You can also select **Affected by Orientation** for either the constrained or constraining object. If you do so, the object rotation is included as part of the constraint for that object.

For example, if you select **Affected by Orientation** for the constrained object, its position is affected by the orientation of the constraining object.

The **Affected by Scaling** option operates in a similar manner for scaling.

5. When you transform the constraining object, the constrained object’s center remains aligned to it.

**To constrain an object by position using constraint compensation**

This method lets you keep the original distance between the constrained object and its constrainer.

1. Position the object to be constrained at the required distance from the constraining object.

2. Click the **CnsComp** button in the Constrain panel to set the offset (see **Setting Offsets with Compensation** on page 320 for more information).

3. Choose **Constrain > Position** and pick the object that is to act as the constraint.

   The offset distance value is displayed in the constrained object’s X, Y, Z position values. You can adjust the value here.

4. Deselect the **CnsComp** button.

5. Translate the constraining object. The constrained object also moves, maintaining the distance you set.

**Pinning an Object in Position**

Instead of having to create an object and then set up a position constraint between it and the object you want to constrain, you can easily pin an object in place with just one command.

**To pin an object**

1. Select the object you want to pin.

2. Choose **Constrain > Pin Object into Position**.

   This creates a “pinning” object in the form of a small implicit sphere (called TEMP_PIN) in the location of the selected object. The selected object is then position-constrained to the pin object.

**To undo the pinning**

- Select the constrained object and choose **Constrain > Pin Object into Position** again to deactivate it.

   This deletes the TEMP_PIN object and removes the position constraint.
Orientation (Rotation) Constraints

You can constrain the rotation values of one or more selected objects to another object without changing the location of the constrained objects. A rotation performed on the constraining object is also performed on the constrained objects. For example, you can make a group of people’s heads all move in sync by animating one head and then constraining the rotation of all the others to this head.

To constrain an object’s orientation

1. Select one or more objects to be constrained.

2. Choose Constrain > Orientation and pick the object that is to act as a constraint.

3. In the Orientation constraint property editor, you can set a Rotation Offset between the centers of the constrained and constraining objects by entering X, Y, and Z values in degrees. See Creating Offsets between Constrained and Constraining Objects on page 318 for more information.

4. Select the constraining object and rotate it as desired. The constrained objects also rotate.
You can set rotation limits for inverse kinematic (IK) chains in its Kinematic Joint property editor—see Setting Rotation Limits on page 165 in the Character Animation guide.
Direction Constraints

You can constrain an axis of one or more selected objects to another object without changing the location of the constrained objects. The aligned axis (the X axis by default) of each constrained object always remains facing the center of the constraining object, thus affecting the orientation of the constrained object.

To constrain an object by direction

1. Select the object to be constrained.
2. Choose Constrain > Direction and pick the object that is to act as a target (constraining object). The constrained object rotates so that its X axis (default axis) faces the constraining object’s center.
3. In the Direction constraint property editor, use the Align Axis controls to specify the axis to point along—X is the default. If two or more values are non-zero, the object points along the resulting vector.

   - You can also add a Target Offset on any axis of the constrained object. This is a rotational offset in degrees from the constraining (target) object (see Creating Offsets between Constrained and Constraining Objects on page 318 for more information).

   - You can also set the Up Vector controls to control the “roll” of the constrained object (see Up-vector Constraints on page 312 for more information).
4. Translate either the constraining object or the constrained object:

- If you translate the constraining object, the constrained object rotates so that its specified axis always faces the constraining object, like a sunflower facing the sun.

- If you translate the constrained object, its specified axis continues to face the constraining object from its new location like a geosynchronous satellite; that is, it is always pointing toward the same location on earth as it follows the earth’s orbit.
Distance Constraints

You can constrain an object so that its center maintains a constant distance from that of the constraining object.

The centers of the smaller planes maintain a constant distance from that of the constraining object (the large plane).

Use either of the following methods to set a distance constraint:

To constrain an object by distance using the property editor

1. Select the object to be constrained.
2. Choose Constrain > Distance and pick the object that is to act as a constraint.
3. In the Distance constraint property editor, set the Distance value, which is a position offset in Softimage units.

You can also set position offsets on any axis of the constrained object using the X, Y, Z sliders in the property editor. See Creating Offsets between Constrained and Constraining Objects on page 318 for more information.
Chapter 10 • Animating with Constraints

4. Translate the constraining object. The constrained object also moves, maintaining the distance you set.

or

To constrain an object by distance using constraint compensation

This method lets you keep the original distance between the constrained object and its constrainer.

1. Position the object to be constrained at the required distance from the constraining object.

2. Click the CnsComp button in the Constrain panel to set the distance offset (see Creating Offsets between Constrained and Constraining Objects on page 318 for more information).

3. Choose Constrain > Distance and pick the object that is to act as the constraint.

   The offset distance value is set for the Distance slider. You can adjust the value here.

4. Deselect the CnsComp button.

5. Translate the constraining object. The constrained object also moves, maintaining the distance you set.
Scaling Constraints

You can constrain the scaling values of one or more selected objects so that they are the same as the scaling values of a constraining object. The **Constrain > Scaling** command is useful for scaling all objects in a hierarchy at once and in accordance to the scaling changes applied to one object (on one or all axes).

To constrain an object's scaling

1. Select the objects to be constrained.
2. Choose **Constrain > Scaling** and pick the object that is to act as a constraint.
3. In the Scaling constraint property editor, you can set a size offset on any axis of the constrained object using the **Scaling Offset X, Y, Z** sliders. See *Creating Offsets between Constrained and Constraining Objects* on page 318 for more information.
4. Scale the constraining object.

The objects constrained to it are also scaled, using the same scaling values as the constraining object.

For more information on scaling in general, see *Transformations* on page 27 in the Transformations guide.
Pose Constraints

The pose constraint lets you constrain all three transformation types (scaling, orientation, and position) using only one constraint. The combination of these three is known as an object’s pose. As well, in the Pose constraint’s property editor, you can selectively constrain any combination of the transformation types, such as constraining only the object's orientation and position, but not its scaling.

This constraint also lets you control the manipulation per transformation type over the hierarchy, which involves making sure two objects always seem to remain connected even if translation, rotation, or scaling is applied to the parent. This involves selective “compensation”: for example, when you rotate or scale the parent object, the rotation or scaling on the child is compensated, but not its translation.

To constrain an object to a pose

1. Select the objects to be constrained.
2. Choose Constrain > Pose and pick the object that is to act as a constraint.
3. In the Pose constraint property editor, you can set an offset for any transformation on any axis of the constrained object using the Scaling, Rotation, and Position X, Y, Z sliders.

   See Creating Offsets between Constrained and Constraining Objects on page 318 for more information.

4. On the Options page, you can activate or deactivate the Position (translation), Orientation (rotation), and Scaling constraints individually. For example, if you deselect the Orientation option, the constrained object is not constrained to the orientation of the constraining object.

5. Transform the constraining object.

   The objects constrained to it are also scaled, rotated, and translated using the same values as the constraining object, depending on which options (Position, Orientation, Scaling) you have active on the Options page.
Symmetry Constraints

The Symmetry constraint automatically mirrors the three transformation types (scaling, orientation, and position) of two objects constrained to each other. You can also use an additional constraint object that acts as a local mirroring reference for the two constrained objects.

Constraining in symmetry is something that you will likely want to do with characters, due to the nature of the human body. For example, Jaiqua’s Tai Chi moves on her left arm's effector are constrained in translation to her right arm’s effector. Her arm movements are mirrored across the YZ plane, in this case.

You can also pick a reference object across which two objects can be constrained in symmetry. The reference object’s XY, YZ, or XZ local plane (whatever you choose in the Symmetry property editor) is the one used for the symmetry for the two objects. This is useful if you want to have a local symmetry constraint across a part of the body, such as the sternum, that moves with the character. If you constrain the arms in symmetry to the sternum, both arms are always symmetrical across it. When manipulating the plane, the constrained objects follow its translation, rotation, and scaling.

Although the Symmetry constraint gets assigned from object A to object B, it is bidirectional. This means that you can manipulate or animate either the constraining or the constrained object. For example, if object A is manipulated or animated, it’s object B that follows in symmetry relative to A; if object B is manipulated or animated, it’s object A that follows. However, if both objects are animated, the constraining object has precedence over the constrained one.

If you’re using the symmetry constraint with bones, the constraining bone must be animated for the constraint to work properly.

To constrain an object symmetrically to another or to a reference object
1. Select the object to be constrained.
2. Choose Constrain > Symmetry and pick the constraining object:
   - If you want to use a reference object, pick it now.
   or
   - Right-click to end picking and select one of the global coordinate planes for symmetry in the following property editor.

3. In the Symmetry constraint property editor, select the global Plane of symmetry across which you want to mirror the animation: XY, YZ, or XZ.

4. Select which transformations you want to use for symmetry by selecting the Active option for Scaling, Orientation (rotation), or Position (translation).

For example, if you deselect Scaling, the constrained object's size is not constrained symmetrically to the constraining object's size.

You can also set an offset for any transformation on any axis of the constrained object using the Scaling, Rotation, and Position X, Y, Z sliders. This can be useful to have the constrained objects not be completely symmetrical, such as having an offset for only the translation in Y.

See Creating Offsets between Constrained and Constraining Objects on page 318 for more information.
Constraints between Points

You can constrain objects between two, three, or any number of points (object centers).

Constraining between Two Points

You can position an object to be constrained between two other objects. By default, the constrained object’s X axis is oriented along the vector that goes from the first-selected constraint to the second one. You can, however, select the object to be constrained by its Y or Z axis.

This constraint is useful for keeping an object between two points pointing along their direction, such as a muscle between two arm joints.

It also provides an easy way to convert two positions into an orientation for motion capture. When you have motion curve data without orientation information, you can infer rotations from position data by using this constraint.

To constrain an object between two points

1. Select the object to be constrained.

2. Choose Constrain > 2 Points and pick the two constraining objects.

   The order in which you select your constraining objects is important to get the correct orientation of the constrained object.

3. In the Two Points constraint property editor, you can set the Distance Percentage, which is the relative distance of the constrained object between its constraining objects.

Plane constrained midway between two points (windsocks)

When the windsocks are repositioned, the plane’s position is updated as well.
A value of 50 (default) positions the constrained object centrally between the two constraining objects. A value of less than 50 positions the constrained object closer to the first constraining object; and a value of more than 50 positions the constrained object closer to the second constraining object.

- You can set position offsets on any axis of the constrained object using the X, Y, Z sliders in the property editor. See Creating Offsets between Constrained and Constraining Objects on page 318 for more information.
- You can set tangency (see Tangency Constraints on page 316) and up vector direction (see Up-vector Constraints on page 312) constraints for the constrained object. By default its X axis is repositioned in the direction of the second constraining object’s center.

4. When you move the constraining objects (one or both), the constrained object readjusts to keep between them.

**Constraining an Object within Three Points**

You can position an object to be constrained within three constraining objects or nulls in a type of invisible triangle. By default, this orients the constrained object’s Y axis to the normal of the plane defined by the three constraining objects. Its X axis is constrained toward the center of the first constraint object you pick.

This constraint is useful for converting three positions from motion capture data into an orientation. When you have motion curve data for positions only, such as three points on the hips and trunk of an actor, you can infer the rotation of its body by using this constraint.

**To constrain an object among three points**

1. Select the object to be constrained.
2. Choose Constrain > 3 Points and pick the three constraining objects.
The order in which you select your constraining objects is important so you can get the correct orientation for the constrained object. By default, the first constraint defines the X axis and the two remaining constraints together define the orientation of the plane for the Y and Z axis.

3. In the Three Points constraint property editor, you can set the U/V/W %age, which is the U, V, and W coordinates to set the relative distance of the constrained object between its constraining objects.

A value of 33.33 for each coordinate (the default) positions the constrained object centrally between the three constraining objects.

A U value of more than 33.33 positions the constrained object closer to the first constraining object; a V value of more than 33.33 positions the constrained object closer to the second constraining object; and a W value of more than 33.33 positions the constrained object closer to the third constraining object.

• You can set position offsets on any axis of the constrained object using the X, Y, Z sliders in the property editor. See Creating Offsets between Constrained and Constraining Objects on page 318 for more information.

• You can set tangency (see Tangency Constraints on page 316) and up vector direction (see Up-vector Constraints on page 312) constraints for the constrained object. By default, its X axis is repositioned in the direction of the first constraining object’s center, and its Y axis is oriented to be perpendicular to the plane formed by the three constraining objects.

4. When you move any of the constraining objects, the constrained object readjusts to keep properly positioned and oriented within its confines.

Constraining an Object within Multiple (N) Points

You can position an object to be constrained centrally within multiple constraining objects or nulls. This constraint calculates the average position of all the points and applies the resulting value as a constraint (a single “pivot” point).

Use blending to have more control when you have many constraining objects and you want to adjust the effect of each one. See Blending Constraints on page 321 for information.

To constrain an object among multiple points

1. Select the object to be constrained.

2. Choose Constrain > N Points and pick all the constraining objects you require.

The order in which you select your constraining objects is important so that you can get the correct orientation of the constrained object.
Object to Cluster Constraints

You can constrain an object to the center of a cluster using the Constrain > Object to Cluster command. This is useful for tracking the position of components as an object deforms.

Looking for the Cluster to Object constraint as in SOFTIMAGE|3D? You can use cluster centers in XSI! When you create a cluster, you can automatically create a cluster center or add it later using the Deform > Cluster Center command in the Model or Animate toolbar. For information on cluster centers, see Cluster Centers on page 133 of the Modeling and Deformation Basics guide.
Object to Cluster Constraints

To constrain an object to a cluster

If you want to constrain an object to an existing cluster, select that cluster first, isolate it in the explorer (press e), and lock the explorer. This makes it easy to pick the cluster after you choose the constraint command.

1. Select the object to be constrained.

2. Choose Constrain > Object to Cluster.

3. Do one of the following:
   - Pick an existing cluster.
   
   or

   - Pick a geometric object, then pick some components or a cluster on it.
     
     If you pick components, a cluster is created automatically.
     
     If you pick multiple component types, you are prompted to confirm which component type to use. You can avoid this prompt by specifying a component selection filter first.

     The object is constrained to the geometric center of the selected components.

4. In the Object to Cluster constraint property editor, you can also do the following to set up the constraint:

   • Set position offsets on any axis of the constrained object using the X, Y, Z sliders in the property editor. See Creating Offsets between Constrained and Constraining Objects on page 318 for more information.

   • Set tangency constraints (see Tangency Constraints on page 316) and normal (up vector) constraints (see Up-vector Constraints on page 312) to control the orientation of the constrained object.
Surface and Curve Constraints

You can constrain the displacement of an object based on the UV values of the constraining NURBS surface object or the U value of the constraining NURBS curve. Much like a surface or curve-based path, the surface and curve constraints orient the Y axis of the constrained object according to the normal value of the NURBS surface’s U/V parameter values, or the NURBS curve at its U parameter value. This allows the constrained object to closely follow the surface or curve.

The surface and curve constraints are useful for keeping an object on a surface or curve—usually when the surface or curve is deformed, such as a ship sailing over a section of choppy water or a car driving along a bumpy road.

For more information in general about NURBS surfaces and curves, see Surface and Curve Modeling on page 1.

The Curve (Param) constraint described in this section is not the same as the Curve (Path) constraint, which is used for path animation that is based on the percentage of the path that the constrained object has traveled. For information on constraining an object to a curve for path animation, see Constraining an Object to a Curve (Path) on page 272.

Constraining to a Surface

To constrain an object to a NURBS surface

1. Select the object to be constrained.
2. Choose Constrain > Surface (Param).
3. Select the constraining object, which must be a NURBS surface object.
4. In the Surface Constraint property editor, click the Normal tab and select Active. Make sure the Y axis is set to 1 in the Affected Axis controls below.

The constrained object’s Y axis is repositioned in the direction of the surface normals of the constraining object.
If necessary, click the eye icon in a viewport and choose Normals to see the object’s normals. If the normals are pointing in the wrong direction, select the surface and choose Modify > Surface > Inverse in the Model toolbar.

5. If you like, animate the constrained object over the surface using the U/V Location sliders.

**Constraining to a Curve**

To constrain an object to a NURBS curve

1. Select the object to be constrained.
2. Choose Constrain > Curve (Param).
3. Select the constraining object, which must be a NURBS curve.
4. In the Curve Constraint property editor, click the Up Vector tab and select Active. Make sure the Y axis is set to 1 in the Affected Axis controls below. The constrained object’s Y axis is repositioned in the direction of the normals of the constraining curve.
   
   Also on the Up Vector page, you can select the Point Along Roll Frame option to “lock” the constrained object to the curve to avoid it flipping.
5. If you like, animate the constrained object along the curve using the U Location slider.
Bounding Volume and Bounding Plane Constraints

The bounding volume constraint makes sure that one or more objects remain within, outside, or on the surface of its constraining object’s boundaries.

In a similar way, the bounding plane constraint constrains an object to the plane of a constraining object.

To apply a bounding-volume constraint

You can use only an implicit cube or sphere as the constraining object for the bounding volume constraint.

1. Select the object to be constrained.
2. Choose Constrain > Bounding Volume.
3. Select the implicit cube or sphere that is to act as a constraint.
4. In the Bounding Volume Constraint property editor, select the bounding type from the Bounding Type list:
   - Inside—the constrained object remains inside the constraining object’s boundary limits.
   - Outside—the constrained object remains outside the constraining object’s boundary limits.
- **On Surface**—the constrained object remains constrained to the constraining object’s outer surface.

- **Monitor Only**—the constrained object exhibits no constraint behavior. If you have selected Constraint Information in the viewport’s Visibility Setting property editor (click the eye icon and choose Visibility Options in the viewport), only the distance between the centers of the constrained and constraining objects is displayed.

5. In the **Constraint Shape Type** list, select whether the object is constrained within the bounding volume by its points, by its surface, or by bounding box points/surface.

6. Transform either a constrained object or the constraining object.

**To use the bounding-plane constraint**

A Bounding Plane constrains an object to the positive Y space defined by another object’s center.

1. Select the object to be constrained.

2. Choose **Constrain > Bounding Plane**.

3. Select the object to act as a constraint.

4. Transform either a constrained object or the constraining object. By default, the constrained object remains confined to one side of the scene as defined by the position and orientation of the constraining object.
Chapter 10 • Animating with Constraints

Up-vector Constraints

The up-vector constraint makes the Y axis of a constrained object point to a constraining (control) object or a specific point. For example, by constraining a camera's up vector to a control object, you can control the camera's up direction to prevent it from flipping when it reaches a vertical axis with the interest.

About the Up-vector Constraint

Unlike the other “real” constraints found in the Constrain menu, the up-vector constraint is a property (actually a low-level expression that determines which direction the object should point) of these other constraints: the Direction, Position, Surface, 2-Point, 3-Point, or Curve (Path). The up-vector constraint cannot exist on its own: it only works only if one of these constraints is already applied to an object.

Because the up-vector constraint is not a real constraint, you cannot remove it as you do other constraints (see Removing an Up-Vector Constraint on page 315 for how to remove an up-vector constraint). However, you can view it as any other constraint in a 3D view or the schematic view (see Viewing Constraints and Their Information on page 288).

When you import an object from SOFTIMAGE|3D that has built-in up-vector constraints, the Up Vector option for that object will be activated in XSI. This affects the Trajectory, Curve (Path), 2-Point, and Direction constraints.
Constraining the Up Vector to an Object

A typical way to determine an object’s up-vector direction is to constrain it to another (control) object. You can constrain the up vector using a menu command or by using the up-vector controls in the constraint’s property editor. The controls in the property editor allow you to easily see and change the up-vector values and relationships.

To constrain the up vector to an object using a menu command

1. Create an object that you will use to control the up vector of the constrained object.
2. Apply one of these constraints to the object whose up vector you want to constrain: Constrain > Direction, Position, Surface, 2-Point, 3-Point, or Curve (Path). Only these constraints have up-vector controls.
3. Do one of the following:
   - Select the constrained object.
   - If the constrained object has more than one constraint with an up-vector property (any of the constraints listed in the previous step), select which constraint you want to use. You can do this in the explorer (constraints are listed in the object’s Kinematics > Constraints folder).
4. Choose Constrain > Up Vector and pick the constraining (control) object.
   The constrained object’s Y axis is now constrained to this object.

To constrain the up vector to an object using the constraint’s property editor

1. Follow steps 1 and 2 as above.
2. In the constrained object’s constraint property editor (Direction, Position, Surface, 2-Point, 3-Point, or Path), click the Up Vector tab and select the Active option to activate the up-vector constraint.
3. Then click one of these buttons on the Up Vector page according to what you want to do:

- Click **Select** to select the current up-vector control object.
- Click **Pick New** and pick a different constraining object (this also activates the Active up vector option if not already active).
- Click **Create New** to create a little implicit cube (named upvtctrl) and use that as the constraining object. It is placed according to the Affected Axis direction (see the next step).

![Diagram of Up Vector page]

4. In the Affected Axis area, select which axis of the constrained object is to be the one used as the up vector (Y is the default). You can either click the axis buttons (positive and negative X, Y, Z) or enter 1 or -1 in the axis text boxes.

Remember that many constraints with up-vector properties already constrain the X axis; for example, it doesn’t make sense to set the up-vector axis to be X for a direction constraint because that constraint already uses the X axis.

You will see the position values of the control object in the Point At X, Y, Z settings, but you won’t be able to move that object using those settings: they serve only to indicate its position.
Constraining the Up Vector without an Object

While it is common to constrain an object’s up vector to a control object, you can simply make the object’s up vector always point in a certain direction using the controls on the Up Vector page. This is known as a static up vector.

To create a static up-vector constraint

1. Select an object and apply any of these constraints which have an up-vector property: Constrain > Direction, Position, Surface, 2-Point, 3-Point, or Curve (Path).

2. In the constraint’s property editor, click the Up Vector tab and select the Active option.

3. In the Affected Axis area, select which axis of the constrained object is to be the one used as the up vector (Y is the default). You can either click the axis buttons (positive and negative X, Y, Z) or enter 1 or -1 in the axis text boxes.

4. Set the Point At X, Y, and Z values to specify a position at which the constrained object’s up vector is to point.

Controlling Flipping with Path Animation

In the Path constraint property editor (see Setting Objects on Paths on page 268), the Point along Path Roll Frame Normal on the Up Vector page constrains the up-vector of the constrained object to the roll-frame normal of the path curve instead of constraining it to a separate control object or direction (as defined by the Point At options on the Up Vector page). The roll frame normal is a frame based only on the moving tangent of a path curve.

When you select this option, the Point At options become unavailable.

Because this option defines the up-vector direction independently from a separate control object or direction, this can help prevent the constrained object from flipping in cases where the trajectories of the path curve and the up-vector control object would cross.

Removing an Up-Vector Constraint

The up-vector constraint is not the same as other “real” constraints, so you cannot remove it in the same way (see Removing (Relaxing) Constraints on page 325). However, you can remove it in the two ways described here.

To remove an up-vector constraint

• Do one of the following:
  - Open the constraint property editor of the constrained object and click the Remove button on the Up Vector page to delete the up-vector control object. You must then deactivate the up-vector constraint.
  or
  - Delete the up-vector control object. You must then also deactivate the up-vector constraint.
Chapter 10  •  Animating with Constraints

**Tangency Constraints**

The tangency constraint makes the X axis of a constrained object tangent to a direction or curve. This is particularly useful for having an object follow a path's direction, such as a roller coaster car following the tracks, or pitching an object (especially a camera) on a path.

Tangency is not a “real” constraint per se, but is available as a property (actually a low-level expression that determines which direction the object should point) of these other constraints: the Surface, 3-Point, Curve (Path), and Object to Cluster.

Because the tangency constraint is not a real constraint, you cannot view it as you do other constraints in a 3D view or the schematic view.

The plane in the top figure is animated on a path but is not constrained to be tangent to the path. It remains oriented this way, regardless of its position on the path.

In the bottom figure, the plane’s Y axis has been constrained to be **tangent** to the path. Note that the nose of the constrained plane does not always rest on the path because the constraint uses the object’s center to set tangency.
To set a tangency constraint

1. Select an object and apply any of these constraints which have a tangency property: Constrain > Surface, 3-Point, Curve (Path), or Object to Cluster.
2. In the constraint’s property editor, click the Tangency tab and select the Active option.
3. Use the Axis to Align controls to specify the axis to point along—X is the default. You can either click the axis buttons (positive and negative X, Y, Z) or enter 1 or -1 in the axis text boxes.

If two or more values are non-zero, the constrained object points along the resulting vector.

Tangency constraints are not calculated correctly when you import a scene from SOFTIMAGE|3D.
Creating Offsets between Constrained and Constraining Objects

When you constrain an object to another, you often need to offset it in some way from the constraining object. Depending on the type of constraint, this could be an offset in position, orientation, or scaling. For example, if you position-constrain one object to another without an offset, both objects end up sharing the same coordinate points and seem to be on top of one another.

You can set offsets between the constrained and constraining objects in most constraints’ property editors. For example, you can offset the constrained object’s rotation for an Orientation constraint, or offset the constrained object’s position in relation to its target object’s location for a Distance constraint.

As well, when you set an offset, you can set the coupling behavior between the constrained and constraining objects to determine how closely the constrained object follows its constrainer’s behavior (see Setting the Coupling Behavior between Objects on page 319).

To set an offset interactively, you can use the Constrain > Constraint Compensation command or the CnsComp button, both found on the Constrain panel. With compensation, you can have the constrained object be interactively offset from the constraining object and animate it independently while still keeping the constraint.

Setting Offsets

With almost all types of constraints, you can set offsets using the controls in their property editors. The offset is set between the centers of the constrained and constraining objects on any axis.
Creating Offsets between Constrained and Constraining Objects

**To set the offset**

- Open a constraint’s property editor and set the values for the X, Y, Z sliders for either the Constraining Object or Constrained Object.

![Image showing how to set the offset.](image)

**Setting the Coupling Behavior between Objects**

When you create an offset for a constraint, you can set the coupling behavior between constrained objects. Although this may sound romantic, the term actually has its roots in the mechanical world and refers to how mechanical parts, as well as many other things, are connected together. For constrained objects, it likewise means how they are connected, more in how they move in relation to each other.

Coupling determines how the constrained object should react when the constraining object is manipulated. When the offsets for both the constrained and constraining objects are set to zero (off), the coupling behavior settings have no effect.

**Soft or Rigid?**

Coupling behavior is available in any constraint property editor that allows you to create offsets for the constrained object. Coupling can be either **soft** or **rigid**, depending on how closely you want the constrained object’s movement to resemble that of its constrainer.

- **Soft** is when objects are allowed to rotate somewhat independently of one another, much like a ball joint. Soft couplings are represented by a small circle in the centers of the constrained and constraining objects.
- **Rigid** is when the constrained object mimics the rotation of the constraining object. Rigid couplings are represented by a small square in the centers of the constraining objects.

The difference between soft and rigid can be thought of in the same way as the difference between a car with a trailer versus a pickup truck. As with soft coupling, the trailer follows the car but still has a limited range of motion; as with rigid, the truck bed is literally welded to the truck.
Setting Offsets with Compensation

To set an offset interactively, you can use the Constrain > Constraint Compensation command or the CnsComp button, both found on the Constrain panel.

You cannot usually edit a constrained object’s parameters because they are subordinate to the constraining object. With compensation, however, you can temporarily override the constraint and modify the constrained parameters (such as position, scaling, or rotation). This means that you can animate the constrained object independently while still keeping the constraint.

When you save keys on compensation, it mirrors the keys “below” on the local transformations; this way, even if you remove the constraint, the offset animation remains the same. You can animate the compensation distance between objects using the CnsComp button and setting keys.

Example

The process of creating an offset between two objects can be easily explained using a simple example of a cylinder and a cone. Let’s say you want to constrain a cylinder to a cone using the Position constraint, but you introduce a slight offset to the constrained cylinder.

1. Create a cone and a cylinder.
2. Select the cone—this will be the object that is constrained to the cylinder.
3. Move the cone to where you would like it to be in relation to the cylinder.
4. Choose Constrain > Compensation or click the CnsComp button in the Constrain panel. This will keep the current position of the cone (soon-to-be constrained object) in relation to the cylinder.
5. Choose Constrain > Position from the Constrain panel (see Position Constraints on page 291) and pick the cylinder. This will be the constraining (controlling) object.
   The cone’s position coordinates are now constrained to those of the cylinder.
6. Deactivate compensation by choosing Constraint Compensation or clicking the CnsComp button again.
   When compensation is active, the CnsComp button turns red to indicate its state. This is because if you forget to deactivate compensation, you can continue to animate in this manner, which is not usually desirable!
7. Now move the cylinder. You will find that the constrained cone maintains the offset.

After you have created a constraint between two or more objects, you can modify their offsets at any time in the constraint’s property editor (see Setting Offsets on page 318).

You could also apply the constraint first and then activate compensation and position the constrained object to create the offset.
Blending Constraints

You can blend multiple constraints on an object with each other, as well as constraints with other animation. When you blend the constraints, you are actually blending the weight (or "strength") of one against the others or even against fcurve or expression animation on the same parameters. And, of course, you can animate the blending to have it change over time.

Blending is done in the order in which you applied the constraints, from the first-applied constraint to the last. Each constraint takes the previous result and gives a new one based on the value you set.

You can see the order of the constraints as well as their blend weight values in a viewport if you have Relations and Relations Info selected (see Viewing Constraints and Their Information on page 288).

Blending Constraints with Each Other

To blend constraints with each other

1. Select an object that has two or more constraints applied to it. These can be any type of constraints.
2. Open the constraints’ property editors.
3. Set the Blend Weight value in each constraint’s property editor, according to the blend you want it to have with the other constraints.

Set keys for this parameter to animate its weight, if you like.

Each constraint takes the previous result and gives a new one based on the value you set. So to blend from one pose to another they have to set the first constraint blend to 1, and the second to the blend percentage.
For example, you could apply three position constraints to an object, with the first-applied weighted at 1, the second one at 0.5, and last one at 0.333. This keeps the constrained object in the middle of the three constraining objects.

_Cone_ has 3 position constraints:
1. First to _A_ with a blend weight of 1
2. Next to _B_ with a blend weight of 0.5
3. Lastly to _C_ with a blend weight of 0.333

This keeps the cone positioned in the **middle** of the triangle formed by _A_, _B_, and _C_.

To easily average the effect of multiple constraints on an object, enter this formula in the Blend Weight text box: \(\frac{1}{\text{constraint ID}}\).

For example, if you have three position constraints on an object and you want the object to be placed exactly in the center of them:

- Enter 1/1 for the first constraint’s blend weight
- Enter 1/2 for the second constraint
- Enter 1/3 for the third constraint.

The blend weight automatically figures out the average weight based on the previous value.
Blending Constraints with Animation

If your object is animated locally (as opposed to globally) with fcurves, you can blend the effect of one or more constraints with the animation. For example, you can animate an object’s position as usual (keys, expressions, etc.) and apply a position constraint to it. You can then use the **Blend Weight** for the constraint to mix the weight of the constraint against the animation.

If the constrained object has no fcurve/expression animation, and the bottom constraint has a weight that is not 1 (full on), what you see is a lag effect. Since there is nothing for the constraint to blend against, it will blend partially (based on the weight) against whatever the last result was. Depending on the blend weight, the constrained object trails what is constraining its pose.

If you want to blend against any non-constraint animation already there (you can blend global against local, not just against itself), then you can weight the first applied constraint to a value other than 1, and the blend will take into account what’s below and blend that in.

**To blend constraints with local animation**

1. Select an object whose local transformation parameters are animated with keys or expressions.

   ![Blend weight for this position constraint is set to 1: other constraint’s weight is set to 0.](image1)
   ![Blend weights for both constraints at about 0.5.](image2)
   ![Blend weight for this position constraint is set to 1: other constraint’s weight is set to 0.](image3)

2. Apply one or more constraints to it (any type).

3. Open the constraints’ property editors and set the **Blend Weight** value.

   ![Blend weights for both constraints at about 0.5.](image2)

   In the example below, the first bone of the chain has keys set on its rotation (FK). However, its effector is position-constrained to two nulls. The blend weight for each constraint is then animated (one from 0 to 1, the other from 1 to 0) so that the effector goes from one null to the other, with the effect of the FK still on the chain.

   ![Blend weights for both constraints at about 0.5.](image2)
Activating and Deactivating Constraints

You can temporarily deactivate a constraint and reactivate it later. You can do this by accessing the constraint’s property editor using the explorer or the viewport.

You can also use the dopesheet (see Deactivating (Muting) Animation in the Dopesheet on page 261) or the constraint query in the spreadsheet (see Constraint Spreadsheet Query on page 290) to activate or deactivate a constraint.

To activate or deactivate a constraint via a property editor

- Open the constraint’s property editor and toggle its Active option.

To activate or deactivate a constraint via the explorer

- Right-click a constraint’s icon and toggle the Active command.

You cannot activate or deactivate all multiple-selected constraints at one time. Only the constraint on which you right-click is activated or deactivated.

To activate or deactivate all constraints for an object

- Right-click the Constraints folder in the explorer and choose Activate or Deactivate to do so for all constraints in that folder.
Removing (Relaxing) Constraints

As opposed to deactivating constraints, where the constraint is still present but has no effect, you can also remove constraints permanently from objects.

When you remove a constraint, the object that was constrained does not return to its “pre-constrained” state; however, if you undo the creation of the constraint (press Ctrl+z), the object returns to its pre-constrained state.

To remove an up-vector constraint, see Removing an Up-Vector Constraint on page 315.

To remove a constraint

Do one of the following:

- In a viewport with the constraint links visible (see Viewing Constraints and Their Information on page 288) between the constrained and constraining objects, click the link and press Delete.

  or

- Select one or more constrained objects and choose Constrain > Remove Constraint. If the object is constrained to more than one object, you are prompted to pick the object from which you want to remove the constraint.

  or

- In the explorer, go to the constrained object’s properties (see Finding Constraint Elements in the Explorer on page 290), select a particular constraint, and press Delete.

To remove all constraints from an object

- Select one or more constrained objects and do one of the following:

  - Choose Constrain > Remove All Constraints.

    or

  - Choose Animation > Remove Animation > from Objects, Constraints from the Animation panel.

    or

  1. In the explorer, go to the constrained object’s Constraint properties (see Finding Constraint Elements in the Explorer on page 290).

  2. Right-click the Constraints folder and choose Delete to remove all constraints in that folder.
Chapter 10 • Animating with Constraints
Chapter 11  Linking Parameters

A linked parameter is a parameter whose animation is driven by the animation of another parameter. You can link any animatable parameters together—from translation to color—to create some very interesting or unusual animation conditions. For example, you could create a chameleon effect so that when object A approaches object B, it changes color. Basically, if you can animate it, you can link it.

After you link parameters, you set the values that you want the parameters to have relative to each other in certain conditions (when A does this, B does this).

Rotation of the lamp post is linked to the position of the space ship on the path. As the space ship nears the post, the post twists.
Chapter 11 • Linking Parameters

There are three basic ways in which you can link parameters. You can:

- Create simple one-to-one links with one parameter driving one or more other parameters. When you link one parameter to another, a simple relationship is established that makes the value of the linked parameter depend on the value of the driving parameter.

  For information, see Linking to a Single Driving Parameter on page 330.

- Drive a single parameter with the combined animation values of multiple parameters. This allows you to create more complex relationships, where many parameter values are interpolated to create an output value for one parameter.

  For information, see Linking to Multiple Driving Parameters on page 337.

- Drive a single parameter with the complete orientation of an object.

  For information, see Linking to an Object’s Orientation on page 341.

Viewing Linked Parameter Information

Because linked parameters are a special type of expression, you can identify the objects that have linked parameters in the 3D views as well as in the schematic view.

To see linked parameter information in a 3D view

- Click the eye icon and choose Relations in the view.

  If the selected object has a parameter on its transformations that are linked to parameters on other objects, those objects are linked to the selected object by dotted white lines.

  You can click a dotted line to see a list of expressions that involve the linked object. You can identify simple linked parameters by the l_fcv expression, multi-linked parameters by the l_interp expression, and orientation-linked parameters by the l_interpOri expression.
To see link information in the schematic view

As is the case with all expressions:

- If an object has linked parameters on its transformations, the letter E is displayed above it.
- For linked parameters, there is also the label L.
- If Show > Expressions is on in the view, an object with an expression is linked to objects whose parameters are involved by solid green arrows labelled Expr.

You can select these arrows by clicking and dragging slightly—a list of expressions appear.
Chapter 11 • Linking Parameters

Linking to a Single Driving Parameter

When you link one or more parameters to a single driving parameter, you specify that each linked parameter’s value is to be controlled (driven) by the animation of the driving parameter. For example, you can have a venus flytrap plant automatically open its mouth when a path-animated “fly” is at a certain distance from it.

Linked parameters provide you with a quick and easy way to create relationships between parameters without having to work out the mathematical expressions involved. They are especially useful with custom parameters, such as when you create a custom control panel with sliders to control a rig (see Custom and Proxy Parameters on page 119 in the Customization guide).

Linked parameters are similar to expressions (in fact they create a special type of expression: \( l_{fcv} \)), but the value of driving parameter controls the linked parameter as dictated by a function curve rather than by a mathematical formula (see Editing Linked Parameter Fcurves on page 333).

While you usually only link one parameter to one driving parameter, you can link several parameters to the same driving parameter. You can also create multiple sets of “one-to-one” links at the same time. No matter how you create the links, an \( l_{fcv} \) expression is created for each linked parameter. As well, when a parameter is linked, its animation icon displays an “L” to indicate this state.

You can also link objects by creating an “equals (=)” expression—see Creating an Expression Without Writing on page 353.

To link one or more parameters to a single driving parameter

• Do one of the following to open the parameter connection editor:
  • Choose View > Animation > Parameter Connection Editor from the main menu.

You can also first select an object whose parameters you want to be linked to (driven by) another parameter and then open this editor.
Linking to a Single Driving Parameter

- In a property editor or explorer, right-click the animation icon of the parameter you want driven by another and choose Link With.

- Choose Link With or Parameter Connection Editor from the Animation menu on the Animation panel.

You can also press Alt while dragging and dropping one parameter’s animation icon to another to create a link and open the expression editor.

Then follow the steps outlined in this overview:

**Overview of Linking Parameters**

1. Select an object, then select one or more of its parameters in the Driven Target explorer. These are the parameters whose values will be controlled by the driving parameter.

2. Select an object, then select one of its parameter in the Driving Source explorer. This is the parameter whose values will control the linked parameters.

3. Select Link With from the link list.

4. Click the Link button.

5. Set the driving and linked parameters’ values as you want them to be relative to each other, then click Set Relative Values.

Repeat this step for each relative state you want to set.
Chapter 11 • Linking Parameters

Creating Multiple Sets of Linked Parameters

In addition to linking multiple parameters to one driving parameter, you can create multiple sets of linked parameters at the same time. The important thing to know for doing this is that you need to have the same number of linked and driving parameters, and the order in which you select the parameters determines what’s linked to what.

For example, if you select ScIX, ScIY, ScIZ parameters in this order in the Driven Target explorer, then select the Posx, Posy, Posz parameters in that order in the Driving Source explorer, a linked parameter is created for each set (PosX drives ScIX, PosY drives ScIY, etc.).

To create multiple sets of linked parameters at the same time
1. Open the parameter connection editor.
2. Select any number of parameters in the Driven Target explorer.
3. Using the same number of parameters, select the parameters in the Driving Source explorer in the order you want to match up with the selected linked parameters.
4. Select Link With from the link list.
5. Click the Link button.

A link relationship is established between each matching set of linked and driving parameters. An l_fcv expression appears in the Definition text box and the animation icon of each linked parameter displays an L to indicate this.

Setting Relative Values

Relative values are “keys” that map a particular value of the driving parameter to a particular value of the linked parameter. They are similar to regular keys that map a particular value to a particular frame. However, instead of the values being mapped over time, the linked parameter’s values are mapped in relation to the driving parameter’s values.

Relative values set the state of the linked parameters in relation to each other.

Here, the venus flytrap’s mouth is closed when the fly is at this distance.

Now the venus flytrap’s mouth is open when the fly is at this distance.

Finally, the venus flytrap’s mouth closes when the fly is in its mouth.
Linking to a Single Driving Parameter

**To set relative values**

1. Set a value for the driving parameter as you want it to be relative to the linked parameter.

2. Set a value for the linked parameter as you want it to be relative to the current state of the driving parameter.

   To make it easier to do this, click the Lock icon in a property editor to keep many editors open at the same time.

3. Do one of the following to set the relative values of these parameters:
   - Click the **Set Relative Values** button in the parameter connection editor.
   - In a property editor or explorer, right-click the animation icon of the linked parameter and choose **Set Relative Values**.
   - Select the object with the linked parameter and choose **Set Relative Values** from the Animation menu on the Animation panel.

4. Repeat these steps for as many relative states as you want to set.

When you set the relative values, a link fcurve is created that represents the relationship between the linked and driving parameters. You can edit the link fcurve in the animation editor, as described in the next section.

**Editing Linked Parameter Fcurves**

When you set the relative values for linked parameters, a link fcurve is created that represents the relationship between the two parameters. Instead of the values being mapped over time, as is the case with typical fcurves, the linked parameter’s values are mapped in relation to the driving parameter’s values. Each key on the fcurve represents a relative value that you have set.
Chapter 11 • Linking Parameters

To edit the link function curve

• Right-click the animation icon of the linked parameter (it should have a little “L” in it) and choose Animation Editor.

The animation editor opens with the link fcurve selected.

To change the function curve interpolation to linear

Like most function curves, the link curve’s interpolation is spline by default, but you may want to change it to linear to prevent having an ease-in and ease-out for the curve (most fcurves should be at a constant speed).

• Select the function curve and click the Linear Interpolation icon or choose Curves > Linear Interpolation in the animation editor’s toolbar.

For information about editing function curves in general, see Editing Function Curves on page 159.
Example: Linking the Opening Doors to a Missile

This is a simple example showing you how to link parameters from two different objects to one animated parameter. You will link the opening of an underground launch pad's silo doors (rotation Z parameters) to the position of the missile (position in Y) taking off from below.

Create the silo doors and the missile using primitives

1. Working in the Front viewport, create the missile doors by getting two cubes and scaling them so they're flat.
2. Get a cone to represent the head of a missile. Position the missile where you want it to be before the doors start opening.
3. At frame 1, set a key for the missile's current position (translation) in Y.

Link each door's rotation with the missile's position

You want the opening of the doors to be affected by the missile's location. To do this, you will link the doors' rotation on Z with the missile's position in Y.

4. Select both doors and open the parameter connection editor. The door objects are displayed at the top of the Driven Target explorer.
5. Select the RotY parameter for each door (Ctrl+click) in the lower part of the Driven Target explorer.
   Click the lock icon for this explorer to prevent it from changing when you select another object.
6. Select the missile object in a viewport so that it appears at the top of the Driving Source explorer.
7. Select the PosY parameter in the Driving Source explorer.
   Click the lock icon for this explorer to prevent it from updating.
8. Select Link With from the link list at the bottom of the parameter connection editor.
9. Click the Link button.
Each door’s RotZ parameter is now linked to the missile’s PosY parameter. An L_fcv expression is created for each linked parameter and displayed in the Definition text box. The animation icon of each linked parameter displays an “L” icon to indicate this.

Set the relative values

Now that the link between each door’s Z rotation and the missile’s Y translation has been established, you must define the relative values between these parameters.

10. Translate the missile along the Y axis to where you want the door to start opening.

11. Click the Set Relative Values button in the parameter connection editor.

A link fcurve has been created for each linked parameter.

12. Go to another frame in the timeline (such as frame 50) and translate the missile along the Y axis to where you want the door to be completely open. Set a key for this value.

13. Rotate each door in Z to create the widest open position.

14. Click Set Relative Values again.

Et voilà! When you play back the animation, notice how the doors automatically open as the missile moves up in preparation for its launch into space!
Linking to Multiple Driving Parameters

When you link one parameter to many driving parameters, you specify that the linked parameter’s value is to be controlled (driven) a combination of all the driving parameters’ animation values. For example, you may want the combination of all three scaling parameters (such as a balloon getting inflated) to drive only the scale Y parameter for an object (an object gets flattened).

Unlike simple linked parameters, where the link is represented by a function curve, a multi-linked parameter has an interpolator (contained within an l_interp expression) that calculates how all the values of the driving parameters are combined together to come up with a result that drives the linked parameter.

To link one parameter to multiple driving parameters

- Do one of the following to open the parameter connection editor:
  - Choose View > Animation > Parameter Connection Editor from the main menu.
  - In a property editor or the explorer, right-click the animation icon of a parameter that you want to be driven by the others and choose Link with Multi.
  - Choose Link With Multi or Parameter Connection Editor from the Animation menu on the Animation panel.

Follow the steps outlined in this overview:

### Overview of Linking to Multiple Parameters

1. Select an object, then select one of its parameters in the Driven Target explorer. This is the parameter whose values will be controlled by the driving parameters. Click the lock icon to prevent the explorer from changing when you select another object.

2. Select an object, then select two or more of its parameters (Ctrl+click) in the Driving Source explorer. These are the parameters whose interpolated values will control the linked parameter.

3. Select Link With Multi from the link list.

4. Click the Link button. A link relationship is established between the parameters. An `l_interp` expression appears in the Definition text box and the animation icon of the linked parameter displays an “L” to indicate this.

5. Set the driving and linked parameters’ values as you want them to be relative to each other, then click Set Relative Values — see below for more information. Repeat this step for each relative state you want to set.

### Setting Relative Values for Multi-Linked Parameters

When you set the relative values for multi-linked parameters, the relative values are combined together and interpolated to create a single result (output) that drives the linked parameter. While there is no function curve created to edit, you can edit the falloff of the interpolation. The falloff determines at what point the values of the driving parameters stop having an influence on the linked parameter.

For orientation-linked parameters, setting relative values maps the complete orientation (not the individual Euler angle values) of an object to the linked parameter — see Linking to an Object’s Orientation on page 341.

**To set relative values**

1. Set values for all the driving parameters as you want them to be relative to the linked parameter. This is the equivalent to first setting the frame before keying a value.
For orientation-linked parameters, set the object’s orientation as you want it to be relative to the linked parameter.

2. Set a value for the linked parameter as you want it to be relative to the driving parameters’ current states.

   To make it easier to do this, click the Lock icon in a property editor to keep many editors open at the same time.

3. Do one of the following to set these relative values:
   - Click the Set Relative Values button in the parameter connection editor.
   - In a property editor or explorer, right-click the animation icon of the linked parameter and choose Set Relative Values.
   - Select the object with the linked parameter and choose Set Relative Values from the Animation menu on the Animation panel.

4. Repeat these steps for as many relative states as you want to set.

   When you set the relative values, an interpolator (contained within an l_interp expression) is created that calculates how all the values of the driving parameters are combined together to come up with a result that drives the linked parameter.

   You can edit the falloff of this interpolation, as described in the next section.

**Editing the Falloff**

When you set the relative values for multi- and orientation-linked parameters, the values are combined together and interpolated to create a result (output).

The interpolator for multi-linked parameters computes “how far” it is from one of its relative values by using a distance metric; for orientation-linked parameters, the interpolator uses the orientation (or quaternion) distance.

While you can’t actually change the way in which the multiple parameters are combined to create the interpolation, you can change the falloff. The falloff determines at what point the values of the driving parameters stop having an influence on the linked parameter.

In the explorer, you can open the linked parameter’s Expression operator property editor which contains the Interpolator.

**To edit the falloff**

1. In the explorer, expand the linked parameter’s branch.

2. Click the Expression operator’s icon to open the expression’s property editor.
3. On the Interpolator page, adjust the **Falloff** value with the slider.

This value determines “how far” the linked parameter is from one of its relative values by using a distance metric; for orientation-linked parameters, this value is the orientation (or quaternion) distance.

The larger the value, the larger the area in which the parameters have an influence. However, if the Falloff value is too large, it can lead to unpredictable behavior.

To edit the relative values inside the interpolator, you can use these three scripting commands: **GetNumRelativeValues**, **GetRelativeValue**, and **RemoveRelativeValue**.

For information on these commands, click the ? icon in the command bar of the script editor to open the online help.
Linking to an Object’s Orientation

As well as linking a parameter to multiple parameters, you can link a parameter to the orientation of an object. When you link to an object’s orientation, you link to the “complete” orientation of an object that is calculated in quaternion space; that is, you don’t link to the object’s individual Euler X, Y, and Z rotation parameters. See Animating Rotations on page 153 for information about quaternion rotation.

The same interpolation technique used for linking to multiple parameters is also used for linking to an object’s orientation—see Linking to Multiple Driving Parameters on page 337. The interpolator for multi-linked parameters computes “how far” it is from one of its relative values by using a distance metric; for orientation, the interpolator uses the orientation (quaternion) distance.

To link a parameter to an object’s orientation

1. Do one of the following to select the parameters to which you want to link the orientation of an object:
   - In a property editor or the explorer, right-click the animation icon of a parameter and choose Link With Orientation.
   - Select the object and mark the parameters that you want to link and choose Link With Orientation from the Animation menu on the Animation panel.

2. Pick the object whose orientation you want to drive the linked parameter. Right-click to end the picking session.

   After you have set up the link, an expression (l_interpOri(local.rotx, local.rotz)) is set on the linked parameter.

   ! If you change the parameters in this expression, the linking will not work any more.

3. Set the object’s orientation, then set the linked parameter’s value relative to the orientation, as described in Setting Relative Values for Multi-Linked Parameters on page 338.

   Setting relative values maps the object’s complete orientation (not the individual Euler angle values) to the linked parameter’s value.

   Now when you change the orientation of the object, the linked parameter changes as well.

4. After you have set the relative values, you can edit the falloff, as described in Editing the Falloff on page 339.
Chapter 11  •  Linking Parameters

Copying and Removing Link Animation

You can copy link animation in the same way as other low-level animation—see Copying Animation on page 52 and Removing Animation on page 69.

Linked parameters are identified by the little “L” in their animation icon.

Link animation can be duplicated between different parameters for the same object or for different objects. For example, you could copy link animation of the X rotation of one object and paste it onto the Y scaling of another object.

With multi- and orientation-linked parameters, the interpolators are copied as well.

When you duplicate link animation, the function curves are copied but not shared. If you change the relative values of one parameter, the other parameters are not affected.

Tips & Tricks with Linked Parameters

This section provides some information about using linked parameters in specific situations.

Using Color Parameters with Linked Parameters

If you want to link parameters with the individual R, G, B, and A channels (red, green, blue, and alpha), you need to use the explorer instead of a property editor. This is because the RGBA sliders in a property editor consist of the four parameters combined, but they are not individually accessible from there.

Using Linked Parameters on Object Components

You cannot use linked parameters to directly manipulate components like point, polygons, edges, or subsurfaces. However, you can create a cluster, assign the cluster center to a null or other object, and use linked parameters with the object to manipulate the components indirectly.

For more information about cluster centers, see Cluster Centers on page 133 of the Modeling and Deformation Basics guide.
**Using Linked Parameters on Preferred Angles**

If you use a linked parameter to control the preferred angle of a chain bone, the parameter’s values will be overridden if you are using the SOFTIMAGE|3D solver.

For more information, see *Choosing a Solver* on page 149 in the Character Animation guide.
Chapter 11 • Linking Parameters
Chapter 12  Animating with Expressions

Expressions are mathematical formulas that you can use to control any parameter that can be animated, such as translation, rotation, scaling, material, or texture. You can create almost any connection you like between parameters, from simple “A = B” relationships to very complex ones using predefined variables, standard math functions, random number generators, and more. However you use expressions, you will find that they are very powerful because they allow you to animate precisely, right down to the parameter level.

If you’re a novice to math, there are some basic expressions you can perform without having to understand math functions well. In fact, many common expressions contain little or no math—some of them just make one parameter equal to another to create a constraint. In this case, all you have to know are the names of the objects and their appropriate parameters. To get started with expressions, try out the exercises in Jumping into Expressions on page 349.
Once you’re more experienced using the mathematical features in XSI, you can create all sorts of custom setups, like character rigs, animation control systems, and simulations. In fact, a knowledge of expressions is often the key to setting up complex character setups in a simple and more direct way than you could using other methods, such as constraints.

An expression is a string of characters that may include object and parameter names, mathematical operators, and tokens representing functions or constants. For example, to constrain object A’s Y rotation to object B’s X translation, you would set an expression on A.kine.local.roty consisting of the string B.kine.local.posx.

Once there is an expression on a parameter, you can no longer manipulate that parameter interactively. For example, if you apply an expression on an object’s local X position, you can no longer use the Translate tool to move the object in X.

You cannot directly use more than one animation source (for example, keys and an expression) on the same parameter. To combine animation sources, see Expressions and Other Animation on page 367.

**Viewing Expression Information**

You can identify the objects that have expressions on their parameters in the 3D views and in the schematic view. As well, you can use an expression query to get information the expressions on an object’s parameters.

**Viewing in a 3D View**

*To see expression information in a 3D view*

- Click the eye icon on the viewport’s menu bar and make sure that Relations is on:
  - If the selected object has parameters with expressions on its transformations, the label Expr is displayed in white.
  - If the selected object has parameters with expressions that involve other objects, those objects are linked to the selected object by dotted white lines.

You can click the dotted line to see a list of expressions that involve the linked object. You can also click one of the expressions in the list to open it in the expression editor.
Viewing in the Schematic View

To see expression information in the schematic view

- If an object has parameters with expressions on its transformations, the letter E is displayed above it.

- If Show > Expression Links is on, an object with an expression is linked to objects whose parameters are involved by solid green arrows labelled Expr.

You can select these arrows by clicking and dragging slightly—a list of expressions appear. Click an expression in the list to open it in the expression editor.

Animation Icon

A parameter’s animation icon changes from its default green box to any of the conditions shown below on the left to indicate expressions of different types (see The Animation Icon on page 26 for more information):

- With a letter C, the parameter has an expression that is a constant value.
Chapter 12 • Animating with Expressions

- With an = sign, the parameter has an expression that makes the parameter equal to another one (links them).
- With an arrow, the parameter has any type of expression on it other than a constant value or an “equal” link.

Expression Spreadsheet Query

You can use a special expression query in the spreadsheet to show the expressions associated with each parameter of a selected object. This query makes it easy to get an overview of expressions on the parameters, as well as activating/deactivating an expression.

To select the query
1. Select one or more objects that have expressions.
2. Open a spreadsheet (press Alt+3) and choose Query > Expressions.

For more information, see *The Spreadsheet* on page 106 in the Interface and Tools guide.

Selecting Objects with Expressions

When you have a crowded scene, it's sometimes difficult to select only the objects you want. To make it easier to select only objects that are animated with expressions, you can use a special expressions filter from the Filters menu on the Select panel.

To use the selection filters
1. In the Select panel of the main command panel, click the little arrow button to open the Filters menu.
2. Select the Obj w Expression filter.
Jumping into Expressions

To understand expressions better, try the following simple exercises. Once you're comfortable with the process and how to use the expression editor, you can go on to do more complicated expressions and setups, as described in Expression Examples on page 369.

Setting Up a Constant Expression

1. Get a primitive sphere (keep it simple!).
2. Select the sphere and press Ctrl+k (this opens the sphere's Local Transform property editor).
3. Right-click the animation icon (green box) of the local Position Y parameter and choose Set Expression.
4. In the expression editor, enter the value of 5 in the expression pane and click the Apply button, but keep the expression editor open to take a peek at what you just did:
   - At the top of the expression editor is the parameter (sphere.kine.local.posy) that you’ve set to be equal to 5.
   - Notice that the Position Y animation icon has the letter “C” in it to indicate that it’s a constant expression (where the value is a number).
Chapter 12 • Animating with Expressions

- An Expression page is added to the Local Transform property editor so that you can easily edit this expression later, including adding comments.

5. Close the expression editor.

Congratulations, you've just set the simplest kind of expression! The sphere is now fixed at 5 units on its Y axis. You won't be able to translate it anywhere else in this direction, but you can still move it in X and Z.

**Trying it another way**

Here's another way of creating a constant expression:

1. Get another primitive sphere and set its Translation Y value to 5.
2. Press Ctrl+k, right-click the Position Y parameter's animation icon, and choose Set Expression.

Notice the expression editor opens with the value of 5 already in it. You've just created a constant expression the same as in the first exercise.

**Creating a Simple Equivalency Expression**

You can also easily create simple A = B expressions between parameters:

1. Open a new scene and get a primitive sphere and a null.
2. Translate the null in X a little to offset it from the origin and the sphere.
3. Select the sphere and press Ctrl+k to open its Local Transform property editor. Lock this property editor to keep it open (click its keyhole icon).
4. Select the null and press Ctrl+k to open a Local Transform property editor for it.
5. Drag and drop the animation icon for the null's Position X parameter to the sphere's Position X parameter in the other property editor.
- The expression editor opens with the affected parameter is sphere.kine.local.posx. Its expression below is null.kine.local.posx, which means that the sphere takes its posx value from the null's posx value.

- The sphere's animation icon for the Position X parameter now has an equal sign (=) in it to indicate the expression.

6. Test the expression by translating the null in X: the sphere should follow in X.
7. Close the expression editor and the two property editors to keep the scene uncluttered.

**Trying it another way**

Here's another way to create an equivalency expression:

1. Select the sphere, press Ctrl+k, right-click the animation icon for Position Y, and choose Set Expression.
2. In the expression editor, type `null.kine.local.posx` in the white expression panel below and click the Apply button.

Now as the null is translated in X, the sphere moves diagonally, getting both its X and Y translation values from the null's position X value.

To continue with more complex examples, see Expression Examples on page 369.
Displaying the Expression Editor

The expression editor is the main tool in which you write and edit expressions. There are several ways to display the expression editor. Each way does something slightly different.

To open the expression editor

There are several ways:

- To display the expression editor without a target parameter or setting a default expression, choose View > Animation > Expression Editor from the main menu. You can then select a target as described in Selecting or Changing the Target on page 356.
- Select an object and press Alt+9 to open the expression editor in a floating window.
- Create a default expression—see the following section, Creating an Expression Without Writing.
- Edit an existing expression—see Editing an Existing Expression on page 362.
- Open the animation editor and choose Editor > Expression Editor.

The number box shows the value of the expression at the current frame.

The target shows which parameter is controlled by the expression.

The message pane updates as you work, letting you know whether the expression is valid.

Press Ctrl+g to switch between this pane and a graph of the resulting expression values.

The expression pane is a text editor where you enter an expression by typing or choosing items from the menus on the command bar.

- You can hide certain parts of the command bar that are optional. To do so, right-click anywhere in this area and toggle the appropriate section on or off (a check mark means that it's displayed).
- If you have animated either global or local transformations with expressions, an Expression page is added to the Global or Local Transform property editor. See Editing an Existing Expression on page 362 for more information.
Creating Expressions

You can create an expression in different ways, depending on how complex it is and which tools you want to use. You can create simple expressions using the explorer or property editor, as described below, or write more complex expressions using the expression editor, as described in Writing Expressions on page 355.

You cannot create an expression on a parameter that already has other animation, such as a function curve or another expression. You must remove the existing animation first—see Removing Animation on page 69.

However, you can combine expressions and other animation using actions and the animation mixer (see Storing Expressions on page 102 in the Nonlinear Animation guide). Using a similar method, you can also have more than one expression on a parameter—see Multiple Expressions on the Same Parameter on page 373.

Creating an Expression Without Writing

While writing expressions allows you to create more complex expressions, you can also create simple and default expressions easily using these methods.

To create a simple $A = B$ relationship

- In property editors, drag parameter A's animation icon onto parameter B's animation icon. Parameter B's animation icon shows an equal sign in it and its value changes to be equal to Parameter A.

- If there is already animation on the animation icon being dragged, the animation is copied to the other parameter, but no expression is set.

- Press Ctrl when you drag and drop the animation icon to create the equal expression without opening the expression editor.

- In the parameter connection editor (see Linking to a Single Driving Parameter on page 330), select parameter B as the Driven Target and parameter A as the Driving Source. Then select Equal (=) Expression from the link list at the bottom of the editor and click the Link button.
In the explorer, drag the *name* of a parameter and drop it on another parameter's name, as shown on the left. The latter parameter's animation icon shows an equal sign in it.

**To open the expression editor and create a default expression**

Use the *Set Expression* command in one of the following ways. In each case, the default expression is the parameter's current value:

- In the explorer or a property editor, right-click the animation icon of a parameter and choose *Set Expression*.

  or

- Mark a parameter, then choose Create > Parameter > Set Expression on the Animate toolbar or Animation > Set Expression from the Animation panel. You can set the same expression on multiple parameters by marking more than one.

If you want to repeat an expression, such as for the Z rotation of all bones, you can select all the bones, then mark the Z parameter, and choose Animation > Set Expression. This gives you expressions on all the parameters at the same time. It also creates multiple expression operators that you can use to tweak the expressions independently.
Writing Expressions

Once you have the expression editor open, you can write an expression by typing it in the expression pane. To write an expression:

1. Select the target parameter. This step is only necessary if you opened the expression editor from the View menu—see Selecting or Changing the Target on page 356.

2. Enter the expression in the expression pane. For a description of expression syntax, see Expression Syntax on page 356. Enter an expression by typing or choosing items from the menus on the command bar.

3. Validate and apply the expression—see Validating and Applying Expressions on page 361.

To write an expression:

- Click the Function button or press Ctrl+f to add a function or constant at the insertion point (see Entering Functions and Constants on page 358).

- Click the Object button to add an object or parameter name at the insertion point (see Entering Object Names on page 357).

- If you have already typed part of a branch name followed by a period, click the Param button to choose a child node to add at the insertion point (see Entering Parameter Names on page 358).

- On Windows systems, you can cut, copy, and paste from another editor (such as Notepad), as well as drag-and-drop text from programs that support that feature.
Chapter 12 • Animating with Expressions

Resulting Value of an Expression

The value of an expression is always a floating-point value (meaning it can be a number like 12.345 instead of an integer like 12). The expression editor always shows the mathematically correct value. However, some parameters can only accept integers or values in a specific range. In these cases, the value returned by the expression is automatically rounded off or clamped before being used.

Tips for Writing Expressions

Here are a few tips to make writing expressions a little easier:

- Copying, cutting, and pasting use standard keyboard shortcuts (Ctrl+c, Ctrl+x, and Ctrl+v, respectively).
- You can preview the result of an expression as a graph. To switch between displaying the message pane and the graph view, press Ctrl+g or choose View > View Graph in the expression editor.
- If you want to manually edit expressions in a bigger pane, press Ctrl+e or choose View > Text Editor to open another text editor, as described in Using Another Text Editor on page 360.
- While you can’t add comment lines to an expression as you can with scripts, you can add comments in the Expression property editor (see Activating and Deactivating Expressions on page 364) or on the Expressions property page that appears in the Local or Global Transform property editor when you create expressions on transformation parameters (see Editing an Existing Expression on page 362).

Expression Syntax

Here are a few things to remember when creating expressions:

- All object names must be followed by a parameter scripting name.
  The exceptions are the distance functions: Center to Center (ctr_dist), to Camera (ctr_dist_cam), and to Camera Interest (ctr_dist_cam_int).
  When you specify an object name with these functions, remember to use a period at the end of the name.
- Parentheses must always be closed.
- Spacing is irrelevant to how the expression is parsed. Spaces, tabs, and carriage returns make an expression easier to read, but they are ignored by XSI.

Selecting or Changing the Target

If you opened the expression editor with the View > Animation > Expression Editor command on the main menu, you first need to select a target parameter (also called the affected element). You can also type in a definition if there is no target selected.
If you opened the expression editor using the **Set Expression** or **Edit Expression** commands, the target parameter is already selected (whatever parameter was marked when you chose the command). However, you can change the target as described below.

**To select or change the target parameter**

1. Do one of the following:
   - Press Ctrl+t.
   - Click the Browse (...) button in the expression editor.
   - Choose **Edit** > **Change Target** from the expression editor’s command bar.
2. In the explorer that opens, you can navigate by expanding nodes until you see the desired parameter, then click its name.

If the desired parameter is listed nearby in the explorer, it may be quicker to use the **Prev** and **Next** buttons in the expression editor’s command bar. These let you quickly apply expressions one parameter after another, such as the transformation parameters.

Alternatively, if you know the scripting name of the parameter, you can type it directly into the target parameter box. To determine the scripting name of a parameter, modify the parameter in a property editor and check the name that gets logged in the SetValue command in the Command box or Script Editor. You can also activate **View** > **Use Script Names** in an explorer. For more information about commands and scripts in general, see *Scripts* on page 25 in the Customization guide.

**Entering Object Names**

You can enter object names by typing directly in the editing pane or clicking the **Object** button and selecting one from a pop-up explorer. You can expand nodes to select children, properties, and parameters.

You can use the **this** token (type **this** or choose **Functions** > **Node** > **this**) in an expression to have an object reference itself. This is useful when you to use the same expression on many different objects. For example, the following always refers to the local Y position of the object whose parameter is the target:

```
this.kine.local.posy
```

You can use the **this_model** token (type **this_model** or choose **Function** > **Node** > **this_model**) to refer to the current model in an expression. For example, if two models Fluffy and Sparky both contain an object named LeftPaw, you can use **this_model.LeftPaw** to refer to either Fluffy.LeftPaw or Sparky.LeftPaw, depending on the context.
You can enter parameter names by typing their script names directly, by clicking the Param button, or pressing F12. These commands prompt you with a list of possible parameters in context.

For example:

1. Enter the following:
   `Camera_Root`.

2. Press F12 to display a list of possible nodes—choose `kine`.

3. Type `l` (lowercase L) and press F12 again. The node local is automatically filled in (it’s the only available node beginning with l).

4. Type `p` and press F12 again. A list of available nodes that begin with p appears—choose `posx`.

Expressions can accept custom parameter names that use a dash (as in `mypset.in-out`).

XSI supports several tokens representing functions and constants in expressions. You can enter these directly by typing in the editing pane or you can select them from the commands in the Function menu.

For a complete description and syntax of all the functions and constants available, refer to Expression Function Reference in the Online Help for the expression editor (click the ? icon).

For some items you choose, only the first part of the string is entered so you must add the closing parenthesis. For example, if you choose Function > Trigonometry > Sine, the following is entered:

```

```

You then type any expression after it, and need to include a closing parenthesis:

```
sin(Fc * 10 + 5)
```

For other items you choose, dummy strings are entered to help you remember which parameters are required. You must replace the dummy strings with expressions. For example, if you choose Function > Condition, the following is entered:

```
cond(<cond>, <true_expr>, <false_expr>)
```

Replace `<cond>` with a condition to evaluate as true or false, for example:

```
(cone.kine.global.posx <= 0)
```

Replace `<true_expr>` with an expression representing the value to use if the condition is true, for example:

```
0
```
Similarly, replace `<false_expr>` with an expression to use if the condition is false.

- If you are using the camera distance functions To Camera (ctr_dist_cam) or To Camera Interest (ctr_dist_cam_int) and you move the camera or interest, you may need to change the current frame to update the scene.
- The Frame Offset (at_frame) and Time Offset (at_time) functions work only for parameters driven by function curves (same as in SOFTIMAGE|3D).

You can enter operators in an expression directly by typing in the editing pane, or you can select them from the Function > Arithmetic menu.

The following operators are supported for expressions:

<table>
<thead>
<tr>
<th>Operator</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>+</td>
<td>Add</td>
</tr>
<tr>
<td>–</td>
<td>Subtract</td>
</tr>
<tr>
<td>*</td>
<td>Multiply</td>
</tr>
<tr>
<td>/</td>
<td>Divide</td>
</tr>
<tr>
<td>%</td>
<td>Modulo (remainder)</td>
</tr>
</tbody>
</table>

Operators follow the standard of precedence: items enclosed in brackets first, then * and /, followed by + and −.

You can enter the following Boolean operators in a condition, available from the Function > Conditions menu:

<table>
<thead>
<tr>
<th>Operator</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>==</td>
<td>Equal to</td>
</tr>
<tr>
<td>&gt;</td>
<td>Greater than</td>
</tr>
<tr>
<td>&gt;=</td>
<td>Greater than or equal to</td>
</tr>
<tr>
<td>&lt;</td>
<td>Less than</td>
</tr>
<tr>
<td>&lt;=</td>
<td>Less than or equal to</td>
</tr>
<tr>
<td>!=</td>
<td>Not equal to</td>
</tr>
<tr>
<td>&amp;&amp;</td>
<td>AND</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Using Another Text Editor

You can use a different text editor to edit expressions. When you choose View > Text Editor or press Ctrl+e, the current contents of the editing pane open in a separate text editor window. You type the expression in the separate window, then when you close the text editor, you are asked whether you want to update the editing pane of the expression editor.

Normally, the default text editor for your operating system is used. You can specify a different editor using the SI_TXT_EDITOR_ENV environment variable. For more information, see Editing the Environment Script (setenv.bat and xsi_4.2) on page 59 in the Setup and Licensing guide.
Validating and Applying Expressions

After you have entered an expression, you can validate and apply it.

Validation checks whether the contents of the editing pane are a proper expression. It provides a way of checking your expression’s syntax without applying it.

Applying an expression attaches it to the target parameters. Changes you make in the editing pane do not take effect until you apply the expression.

To validate an expression

Click the Validate button on the expression editor’s command bar. A message appears in the message pane, informing you whether the expression is valid or not.

To apply an expression

Once you have entered an expression in the bottom pane, click Apply. The expression is automatically validated first; if it is not valid, a message informs you and the expression is not applied.

Closing the expression editor applies the expression automatically.

Copying and Removing Expressions

You can copy and remove expression animation in the same way as other low-level animation—see Copying Animation on page 52 and Removing Animation on page 69.

Expressions can be duplicated between different parameters for the same object or for different objects. For example, you can copy an expression from the X rotation of one object and paste it onto the Y scaling of another object.

The this and this_model tokens are useful when using the same expression on many different objects—see Entering Object Names on page 357.
Chapter 12 • Animating with Expressions

Editing an Existing Expression

To edit an existing expression

Use one of the following ways:

• In the explorer or a property editor, click the animation icon of a parameter animated with expressions. This opens the expression editor within the animation editor.

or

• In the explorer, expand the parameter whose expression you want to edit so that you see its Expression operator. Click the Expression operator’s icon to open its property editor. As well, you can add comments to the expression here.

or

• If you have animated either global or local transformations with expressions, you can open the appropriate Global or Local Transform property editor.

An Expression page is added to the property editor, which displays the expression so that you can easily edit it. As well, you can add comments to the expression here.

or

• In the explorer or a property editor, right-click the icon of a parameter with an expression and choose Expression Editor.

or

Animation icon of a parameter animated with any of these types of expressions:

- Constant value
- Equal to another parameter
- Any other expression
• Mark a parameter, then choose Create > Parameter > Edit Expression on the Animate toolbar or Animation > Expression Editor from the Animation panel.

If the parameter does not already have an expression, it is set as the target parameter, but no default expression is created automatically.

To open and edit an existing expression from other views

• In the animation editor, from its tree select a parameter animated with expressions (see Using the Animation Tree on page 162).

• In a 3D view, click the eye icon on the viewport’s menu bar and make sure that Relations is on.

If the selected object has an expression relating it to another object, they are joined by a white dotted line labeled Expr. Click the line to see a list of expressions; select an expression from the list to edit it. See Viewing Expression Information on page 346 for an example.

• In the Schematic view, make sure that Show > Expression Links is on.

If there is an expression connecting two objects, it is shown as a green line labeled Expr. Select the line to see a list of expressions; select an expression from the list to edit it. See Viewing Expression Information on page 346 for an example.
Activating and Deactivating Expressions

You can temporarily deactivate an expression and reactivate it later, as well as animate this condition. When deactivated, the value of the driven parameter remains the same as of the last update (as with deactivated constraints).

You can also use the dopesheet (see Deactivating (Muting) Animation in the Dopesheet on page 261) or the expressions spreadsheet query (see Expression Spreadsheet Query on page 348) to activate or deactivate an expression.

To activate or deactivate an expression

Do either of the following procedures:

1. In the explorer, expand the parameter whose expression you want to activate or deactivate so that you see its Expression operator.
2. Click the Expression operator’s icon to open its property editor.
3. Toggle the Active option to activate/deactivate the expression.

or

• If you have animated either global or local transformations with expressions, you can open the appropriate Global or Local Transform property editor.

An Expression page added to the property editor, which displays the expression so that you can easily activate or deactivate it.
Saving and Loading Expressions

You can save the current contents of the expression editing pane as a file then load it back later on other parameters. Expressions are saved as plain text files with an extension of .expr2. By default, they are saved in the Expressions folder of the current project.

As well, you can save and load expression presets.

Expressions are not validated when saving—the contents of the editing pane are written as is to file. Similarly, expressions are not validated when opening—the contents of the editing pane are replaced with the contents of the .expr2 file.

To save the contents of the editing pane in an expression file

Choose one of these commands from the expression editor’s File menu:

- Save—If you have already opened or saved an expression file, it is overwritten on disk with the current contents of the editing pane. Otherwise, you are prompted for a file name as with Save As.

or

- Save As—A browser opens, prompting you for a file name and directory.

To open a saved expression file

Choose File > Open in the expression editor, then use the browser to select the desired .expr2 file.

You cannot open a binary .exp format expression file from SOFTIMAGE|3D directly. Instead, import a scene or model with the expression in it.

To save an expression preset

1. In the explorer, expand the parameter whose expression you want to save as a preset.
2. Click the Expression operator icon to open its property editor.
3. Click the Save button in the expression editor to open the browser.
4. Enter a name and location for the preset.

You can edit the Definition and save numerous versions of this expression as presets.

To load a preset
1. Select the parameter to which you want to apply a preset.
2. Click its Expression operator icon to open its property editor.
3. Click the Load button and select a preset from the browser.

**Importing Expressions from SOFTIMAGE|3D**

When importing scenes and models with expressions from SOFTIMAGE|3D, you should be aware of a slight difference that affects a small number of cases. With the distance functions `ctr_dist`, `ctr_dist_cam`, and `ctr_dist_cam_int`, SOFTIMAGE|3D does not evaluate past the object name. This means that in SOFTIMAGE|3D the following string is valid even if `obj1` is not on a path:

```
ctr_dist(pref.obj1.trans, pref.obj2)
```

However, when the expression is imported, the result is invalid because XSI tries to resolve the path reference. To avoid this and similar problems, change `pref.obj1.trans` to `pref.obj1` before importing.

**Other Differences from SOFTIMAGE|3D**

The following are some other differences between expressions in SOFTIMAGE|3D and XSI:

- The SOFTIMAGE|3D function `has_relation()` is not supported in XSI.
- The modifiers `.L` and `.G` are not used to specify local and global transformations. XSI specifies these parameters with `.local` and `.global` in a context like this:
  
  ```
sphere.kine.local.posx
sphere.kine.global.posx
  ```
- You cannot combine expressions and function curves directly on the same parameter. To do this, you must use actions—see Storing Expressions on page 102 in the Nonlinear Animation guide.
- XSI does not support local variables. If you import a scene from SOFTIMAGE|3D containing local variables, they will be expanded. You can use custom parameters to mimic local variables, as described in Using Custom Parameters as Variables on page 374.
Expressions and Other Animation

Here are some ways in which you can use expressions with other types of animation on the same object or parameter.

Combining Expressions and Constraints

Expressions are evaluated before constraints on the same object. If you use the value of a constrained parameter in an expression, you may not get the desired results.

For example, suppose you want to write an expression to control an object's orientation based on its position; if the object has a path constraint, the orientation is evaluated before the position is updated at each frame so the expression uses stale values. To achieve the desired result, constrain an intermediate object like a null to the path then constrain the object's position to the null. When calculating the new orientation, use the null's position.

It is possible to create dependency loops with constraints, expressions, and scripted operators. As a result, a warning appears in a message box if you apply an expression that introduces a cycle. See Checking for Animation Dependency Cycles on page 66 for more information.

Combining Expressions and Fcurves

While you can't have both fcurves and expressions on the same parameter at the same time, there are some workarounds to doing this.

- If you want to globally clamp an object, you can place an expression on the global transformation parameter, say on the Y Position, and continue happily animating with local fcurves on the object.

- You can have an expression “on top” of an fcurve using clips on the mixer. For example, you can write an expression that clamps the Local Y Position and place it in an action which you drop in the mixer. You can then continue to animate on that channel of animation with fcurves and the mixer will temper the value with the expression within the clip.

Because you can activate or deactivate an expression (see Activating and Deactivating Expressions on page 364), you can have an expression with other animation on the same parameter, but it can’t drive the animation at the same time as other animation.
You can convert an expression to a raw function curve file (.fraw2 format). XSI plots the value of the expression and saves a function curve file with a key at every frame. You can then open the saved .fraw2 file in the animation editor. The original expression applied on the selected parameter is left unchanged.

To convert an expression to a function curve file

1. Make sure that you have applied the expression. XSI plots an expression only if it has been applied; it does not plot the contents of the expression editing pane.

2. In the expression editor, choose File > Save as Fcurve.

3. In the browser that opens, select a directory and file name, then click OK.

For information about opening function curves in the animation editor, see Saving and Loading Function Curves on page 239.
**Expression Examples**

Let the math begin! Here are a few examples of some more advanced expressions.

**Relational Expressions**

At the beginning of this chapter (*Jumping into Expressions* on page 349), you saw how Parameter B can directly take the value of Parameter A. But you can also use the values from parameter A and modify them.

1. Create a sphere and a cube, just to keep things simple.
2. Animate the cube moving back and forth on its X axis.
3. Select the sphere and press Ctrl+k to open its Local Transformation property editor.
4. Right-click the animation icon of the Position X parameter and choose **Set Expression**.
5. Enter the following into the expression editor:
   \[ \text{cube.kine.local.posx } \cdot -1 \]
6. Validate and apply the expression and close the expression editor.

   Notice the expression on its own page inside the Local Transformation property editor. The animation icon for the Position X parameter now has an arrow in it indicating that the parameter is being controlled by a relationship.

7. Play the animation and notice that as the cube moves back and forth on the X axis, the sphere moves the same but in the opposite direction (the -1 value did this).

   If you plotted the Position X animation of the sphere, you would see that its fcurve is the same as the cube’s Position X fcurve, except that it would be flipped vertically.

8. Edit the expression in the property editor to the following:
   \[ \text{cube.kine.local.posx } \cdot -0.5 \]
Notice how the sphere now moves in the opposite direction, but only half as far. If you plotted its Position X fcurve, you would see that it’s upside-down and half as tall as the cube’s Position X fcurve.

By multiplying the value of the cube’s Position X parameter, you’re basically altering the shape of the source fcurve. You can make the fcurve taller by multiplying by numbers greater than 1, or shorter by dividing. Adding and subtracting values to the parameter shifts the resulting fcurve left and right.

9. With the sphere selected, choose Create > Parameter > New Custom Parameter Set from the Animate toolbar and name the set Control.

10. Choose Create > Parameter > New Custom Parameter (or press Shift+p) to create a custom parameter for the set and name it slider.

11. Set the Minimum value range to -10 and Maximum to 10 and leave everything else as a default.

12. Edit the expression for the sphere’s Position X parameter again, but this time as follows:

\[
\text{cube.kine.local.posx} \times \text{sphere.Control.slider}
\]

13. In the explorer, find the Control custom parameter set under the sphere’s node and click its icon to open up its property editor. Inside, there should be the parameter you created (slider).

14. Loop the animation of the cube while dragging the custom parameter’s slider across its range of values.

Instead of multiplying the cube’s Position X parameter by a number like -1 or -0.5, you multiplied it by the slider’s value. You can see that this slider now controls the amplitude of how the sphere translates in X relative to the cube translating in X.
You can imagine that when the slider is set to 1, the sphere’s Position X fcurve matches the cube’s Position X fcurve precisely. As you increase the slider value, the sphere’s Position X fcurve grows taller. As you decrease the value to zero, the fcurve flattens out, only to grow again in the opposite direction as you move the slider to negative values.

15. Change the expression so that instead of multiplying, you add the two parameters together:

```
cube.kine.local.posx + sphere.Control.slider
```

Now as you move the slider, the sphere shifts to the right or left.

This sort of mathematical manipulation of fcurves is exactly what’s taught in math analysis or pre-calculus. Basically, it’s drawing pictures with math equations. A pre-calculus text book can give you all the information you’ll need to know about how to manipulate the shapes of functions using different math operations.

**Conditional Expressions**

If you’re getting into expressions, try this simple exercise using a conditional expression.

1. In a new scene, get a primitive sphere.

2. Get two primitive nulls and translate them so that one is above the sphere and the other is below so that they have noticeably different Y positions.

3. Set an expression on the sphere’s local Position Y parameter:

```
cond( sphere.kine.local.posx > 0, null.kine.local.posy, null1.kine.local.posy )
```

This is a conditional expression, which works according to this format:

```
cond( True or False Condition, This Value if True, This Value if False )
```

If you choose Functions > Conditions > Condition in the expression editor, this formula is added for you in the expression pane so that you don’t have to type it all out.

4. Translate the sphere on its X axis to see how it jumps when the value is greater than or less zero.
Chapter 12 • Animating with Expressions

5. If it's greater than zero, the Position Y value from the first null is used; otherwise, the Position Y value from the second null (null1) is used.

Oscillation Using Cosine

And now for a simple oscillation using the cosine function and the frame rate of the scene's playback.

1. In a new scene, get a primitive sphere.
2. Create a custom parameter set for it called Control.
3. Create two custom parameters for this set, called \texttt{amp} and \texttt{freq}, each with value ranges between -100 and 100.
4. Set the following expression on the sphere's local Position X parameter:
   \[
   \cos( T \times 90 \times \text{sphere.Control.freq} ) \times \text{sphere.Control.amp}
   \]

   You can also find the cosine function in Functions > Trigonometry > Cosine in the expression editor.

   T represents Time and is dependent on the frame rate setting of your playback. If it is set to 30 frames per second, T will equal 1 at frame 30.
5. Set the playback in a loop, play around with the custom \texttt{amp} and \texttt{freq} sliders (set the \texttt{freq} first).
Tips & Tricks with Expressions

This section provides some information about using expressions in specific situations.

Multiple Expressions on the Same Parameter

In XSI, an expression is treated like any fcurve-based animation, meaning you can use the mixer to mix its weight. If you store your existing animation in an action clip, you can write your expression and store that in its own clip as well. Then you can mix the two clips and make sure your expression affects the parts of your animation you want it to.

You may want to have more than one expression driving the same target. The expression editor will not allow you to create an expression on a parameter if there is already another expression driving it.

The following steps provide a workaround using the animation mixer:

1. Create your first expression on a target, then mark the parameter and choose Actions > Store > Marked Parameters - All Sources from the Animate toolbar.

2. Now that you have an action of the expression (and have removed the original expression from the parameter), set the next expression directly on the parameter.

3. In the Animation Mixer, load the action. You now have two expressions on the same parameter.

Using Color Parameters with Expressions

If you want to set expressions on parameters with the individual R, G, B, and A channels (red, green, blue, and alpha), you need to use the explorer instead of a property editor. This is because the RGBA sliders in a property editor consist of the four parameters combined, but they are not individually accessible from there.
Using Custom Parameters as Variables

Local variables are not currently supported in expressions, but you can store values or animation in a custom parameter for the object you are driving and then reference that custom parameter within the expression. You must then change only the custom parameter to update all expressions that reference it.

Using Expressions on Object Components

You cannot use expressions to directly manipulate components like point, polygons, edges, or subsurfaces.

However, you can create a cluster, assign the cluster center to a null or other object, and use expressions on the object to manipulate the components indirectly.

For more information about cluster centers, see Cluster Centers on page 133 of the Modeling and Deformation Basics guide.

Using Expressions on Preferred Angles

If you set an expression on the preferred angle of a bone, the expression’s values will be overridden if you are using the SOFTIMAGE|3D solver. For more information, see Choosing a Solver on page 149 in the Character Animation guide.
Chapter 13  Device Drivers

Device drivers are connections between the device plug-in (such as an electronic keyboard, MIDI board, or mouse) and XSI that get and set the values or execute commands or events. You then use the device to animate the object and capture the information, resulting in function curves.

A device driver contains a collection of channels and can implement a dialog box to configure the device, either through the user interface or the description file.

There are three device drivers provided in XSI for you:

- A general MIDI keyboard driver
- A motion capture mouse driver
- A PC1600 MIDI board driver

If you want to create your own device driver, you can write one using the SDK. See the SDK Plug-in Integration guide on the Documentation CD for more information.

Device drivers are exposed in the object model and can therefore be used in scripting.
**Overview of Using Device Drivers**

This is a basic overview of how to use device drivers with XSI:

1. Make sure the devices you want to use are connected to your computer and the driver installed.

2. Choose **Tools > Devices > Manage Device Drivers** from the Animate toolbar to open the Device Manager.

3. Add a driver to the list—see *Adding and Removing Device Drivers* on page 377.

4. Set up the driver’s channels—see *Setting Up the Device Driver’s Channels* on page 379.

5. Make sure you’re playing the scene in realtime playback—see *Playing Back All Frames or Playing in Real Time* on page 85.

6. Start capturing animation information—see *Capturing Animation Information* on page 381.
Managing the Device Drivers

The Device Manager window displays a list of all devices available in XSI which you can activate or deactivate. You can add drivers to or remove them from this list, as well as access each driver’s channels for editing. You can also start and end a capturing session from here.

To open the Device Manager

- Do one of the following:
  - Choose Tools > Devices > Manage Device Drivers from the Animate toolbar.
  - In the explorer, go to the Project level, expand the Data tree, and click the Device Manager icon.

To add a device driver to the manager

1. In the Device Manager window, click the Add button. The dialog box for selecting a device driver appears.
2. Click on the device you want to add to the manager and click Select. By default, the MIDI Keyboard, Mouse, and PC1600 (MIDI board) device drivers are listed.
3. The Device Driver property editor opens in which you can set up the channels (see Setting Up the Device Driver’s Channels on page 379) and activate the driver (click the Active option).

4. The device driver is displayed in the Device Manager window. Click in its Active cell to activate it there.

To remove a device driver

- Select a device driver from the list in the Device Manager and click the Remove button.

**Activating and Deactivating Device Drivers**

To activate or deactivate all available device drivers

- Do one of the following:
  - Select the Device Manager Active option in the Device Manager window.
  - Choose Tools > Devices > Enable/Disable All Device Drivers from the Animate toolbar.

To activate or deactivate individual device drivers

- Do one of the following:
  - In the Device Manager window, click in each device driver’s Active cell.
  - Open the device driver’s property editor (see next section) and toggle its Active option.

- To reconnect all channels of all active devices that are using the current selection, select Options > Track Selection.
Setting Up the Device Driver's Channels

You can set up each device driver's channel in its property editor. This includes activating a channel, setting an action with a target, and setting the channel's value (possibly including an offset and scale).

You can save presets for device driver settings in its property editor as you can with anything else in XSI—see Creating Presets of Property Settings on page 110 of the Scene Elements guide for more information. This makes it easy to create a library of different channel setups for a particular device, then load them easily from the property editor.

To set up the channels

1. Do one of the following to open a device driver’s property editor:
   - Select a driver from the list in the Device Manager and click the Inspect Selection button. The driver’s property editor appears.
   or
   - In the explorer, go to the Project level, expand the Data > Device Manager tree. Each device that appears in the Device Manager window appears here. Click the icon for the device driver you want to edit so that its property editor appears.

   The devices that XSI recognizes on startup will not appear in the explorer until you add them to the Device Manager.

2. Activate or deactivate a channel by clicking in the channel’s Active cell. A channel is not used if it’s not active here.

3. If you like, offset or scale the channel’s value by entering a value in the Scale or Offset cell.

4. Map a channel’s action to a target by first clicking Action and selecting a type from the list (see the following table).
5. Enter the parameter or command to which you want this action to be connected in the Target text box.

The possible Actions and Target combinations are listed in the following tables:

<table>
<thead>
<tr>
<th>Action Type</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>Nothing</td>
</tr>
<tr>
<td>Drive</td>
<td>Parameter name</td>
</tr>
<tr>
<td>Command</td>
<td>Any XSI command</td>
</tr>
<tr>
<td>Special</td>
<td>Command name in the following table.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Target Command</th>
<th>Does this ...</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACTIVATE,driver_ID</td>
<td>Activates the specified device driver.</td>
</tr>
<tr>
<td>DEACTIVATE,driver_ID</td>
<td>Deactivates the specified device driver.</td>
</tr>
<tr>
<td>CURRENT</td>
<td>Current selection.</td>
</tr>
<tr>
<td>TIMELINE</td>
<td>Drives the timeline.</td>
</tr>
<tr>
<td>TIME_START</td>
<td>Goes to the first frame.</td>
</tr>
<tr>
<td>TIME_END</td>
<td>Goes to the last frame.</td>
</tr>
<tr>
<td>TIME_STEP_FORWARD</td>
<td>Goes to the next frame.</td>
</tr>
<tr>
<td>TIME_STEP_BACK</td>
<td>Goes to the previous frame.</td>
</tr>
<tr>
<td>TIME_STOP</td>
<td>Stop the playback.</td>
</tr>
<tr>
<td>TIME_PLAY_NORMAL</td>
<td>Plays back from current frame (real-time playback).</td>
</tr>
<tr>
<td>TIME_PLAY_FRAME</td>
<td>Plays back from first frame (real-time playback).</td>
</tr>
<tr>
<td>TIME_PLAY_SLIDE</td>
<td>Plays from current frame (all frames).</td>
</tr>
<tr>
<td>TIME_SPECIFY,frame</td>
<td>Goes to the specified frame.</td>
</tr>
</tbody>
</table>
Capturing Animation Information

You can capture information with the device from within the Device Manager window, by using keyboard shortcuts, or by choosing menu commands.

To capture animation information

1. Make sure that the appropriate device driver is active in the Device Manager.
2. Select realtime playback for capturing (see Playing Back All Frames or Playing in Real Time on page 85).
3. Select the object whose animation you want to capture.
4. Do one of the following to start the capturing:
   - In the Device Manager, click the Start button to start the session, which turns on all active device drivers. Press the space bar to start the capture.
   - Press Ctrl+Shift+up-arrow.
   - Choose Tools > Devices > Start Capture from the Animate toolbar.
5. Use the connected device to animate the object.
   For example, if you have the mouse driver active, move the mouse to animate the object according to the channels you set up. Each time the mouse moves, a key is set. No key is set if there is no movement.
6. To end the capturing, do one of the following:
   - In the Device Manager, press the space bar again. Then click the Stop button to stop the session, which turns off all active device drivers.
   - Press Ctrl+Shift+down-arrow.
   - Choose Tools > Devices > Stop Capture from the Animate toolbar.
7. Function curves are created for the parameters that you animated with the device.
   See Chapter 7: Editing Function Curves on page 159 for more information.
Index

Numerics
2-point constraints 303
3-point constraints 304

A
action sources
plotting animation to 59
viewing in fcurve editor 180

Animate toolbar 24
Refer also to Help > XSI Reference
animation
baking into one source 59
capturing in viewport 91
capturing information with devices 381
commands for 24
copying and pasting 52
creating curves from 274
cycles of function curves 223
editing function curves in fcurve editor 160
editing in dopesheet 241
information in scene (info scene) 28
levels of in XSI 21
local and global transforms 142
low-level 21
muting 65
muting function curves 236
muting in dopesheet 261
offsetting 55
on paths 265
order of operations 22
overview 19
playback range in timeline 78
playing 80
plotting 59
previewing in flipbook 94
removing 69
removing static (flat) fcurves 70
reversing 56
scaling 55
setting preferences 28
timing 203
transformations 139
visual cues 157
See also expressions
animation cues 157
animation curves See function curves
animation editor
displaying 160
displaying other editors 161
dopesheet in 242
expressions in 161
fcurve editor in 166
function curves in 159
playback cursor 164
scripted operators in 162
setting preferences for tree 163
timeline 164
tree 162
animation evaluation cycles 66
animation icon
about 26
expressions 362
in property editor 117
Animation menu in Animation panel 25
Animation panel 25
animation trails 49
audio
capturing with animation 93
exporting with flipbook images 99
auto button (autokeying) 122
autokeying
as default mode 121
setting keyframes automatically 121

B
baking
animation into one source 59
blending
constraints with each other or animation 321
Boolean
operators in expressions 359
parameters in dopesheet 261, 263
bounding plane constraint 310
bounding volume constraint 310
C
capturing
animation in viewport 91
animation information with devices 381
animation with audio 93
centers
animating transformations 152
channel box, See keying panel
clusters
constraining objects to center 306
commands
for animation 24
comments
for expressions 362, 364
compensation
offsets between constrained objects 320
components
animating 342, 374
conditional
expressions 371
constant interpolation for function curves 211
constants
in expressions 358
constraints
about 283
activating and deactivating (muting) 324
aligning with normals of curves 308
aligning with normals of objects 308
blending multiple 321
blending with animation 321
bounding-plane 310
bounding-volume 310
cluster to object 306
combining with expressions 367
compensation (offsetting) 320
coupling behavior 319
curve (U value) 308
direction 295
distance 297
elements in explorer 290
evaluation cycles 66
global space 284
linking parameters 327
multiple 284
multiple (n) points 305
muting in dopesheet 261
object to cluster center 306
offsets between objects 318
order of 284
orientation (rotation) 293
pinning an object in place 292
plotting transformation parameters 61
pose (scaling, orientation, and position) 300
position 291
removing 325
scaling 299
schematic view 289
selecting objects animated with 286
selecting the constrained object 287
selecting the constraining object 287
spreadsheets query 290
surface normals 308
symmetry 301
tangency 316
three-point 304
to curve (path) 272	two-point 303
undoing 285
up-vector 312
viewing information 288
continuous rotation
curves 156
plotting 64
converting
expressions to function curves 368
object translation to a path 278
quaternion and Euler rotations 154
raw and standard function curves 237
copying
animation 52
expressions 361
link animation 342
coupling behavior for constraints 319
curve editor See fcurve editor curves
constraining objects to 308
constraining objects to (path) 272
creating from animation 274
See also function curves
custom parameters and expressions 374
CycleChecking command 66
cycles
checking for animation 66
creating for function curves 223
D
deactivating See muting deformations
deformations See muting
deleting
muting in dopesheet 261
 deleting animation 69
key points on function curves 197
dependency cycles between animated objects 66
device drivers
activating for use 378
adding to list 377
capturing animation information 381
function curves created by
capturing 381
managing 377
removing from list 377
setting up channels 379
Device Manager 377
direction constraints 295
distance constraints 297
dopesheet
about 241
aligning tracks with parameters in tree 249
collapsing and expanding tracks 248
copying animation between models and hierarchies 259
cutting and copying regions 133, 256
deactivating (muting) animation 261
deleting keys 137, 253
drawing regions 252
empty regions (no keys) 132, 134, 254, 255, 256
framing 246
hiding tracks 249
in timeline 243
keyframe colors 251
marking parameters 250
moving keyframes 251
moving regions 129, 254
panning 245
pasting regions 256
rippling 127, 253
scaling regions 131, 255
setting preferences 243
summary track 247
tracks 247
zooming 245
driven keys See linked parameters
E
dead frame in timeline 78
equalize Bézier handles See slope handles, equal length
Euler rotations
converting to and from quaternion 154
versus quaternion 153
evaluation cycles for animation 66
explorer
animation elements in 30
constraint elements in 290
path elements in 267
tree in animation editor 162
exporting
flipbook 98
function curves 239
EXPR2 (expression) files 365
expression editor
about 352
in animation editor 161
opening 352
expressions
about 345
activating and deactivating (muting) 364
adding comments to 362, 364
and custom parameters 374
applying 361
Boolean operators 359
combining with constraints 367
combining with function curves 367
conditional 371
constants 358
converting to function curves 368
copying 361
creating 353
editing 362
equal relationships between parameters 353
evaluation cycles 66
examples 369
expression editor 352
files 365
functions 358
functions available Refer to Online Help
has_relation function 366
importing from

SOFTWARE|3D 366
in global or local transform property editor 362, 364
linked parameters l_fcv() 330
linking parameters 327
loading from file 365
math operators 359
multi-linked parameters l_interp() 337
multiple on same parameter 373
muting in dopesheet 261
object names 357
on color parameters 373
on components 374
on preferred angles 374
operators 359
oscillation 372
overview of workflow 349
parameter names 358
presets 365
relational 369
removing 361
returned value 356
saving 365
selecting objects animated with 348
spreadsheet query 348
syntax for writing 356
target (affected element) 356
this token 357
this_model token 357
using other text editors 360
validating 361
viewing as graphs 352
viewing information 346
writing 355
extrapolation
function curves 222
F
falloff
for multi-linked and orientation parameters 339
fast playback 86
fcurve editor 166
command bar 166
default selection tool 187
editing fcurves in 159
framing 183
grid and rulers for fcurve graph 171
metacurve region 180
opening with specific fcurves 168
panning 182
setting preferences 166
zooming 181, 182
fcurves See function curves
file types .expr2 365
.fraw 239
.fraw2 239, 368
fitting
function curves 234
flip.exe (standalone flipbook) 94
flipbooks
capturing audio for 93
capturing images for 91
exporting 98
exporting audio with images 99
launching after capturing 92
playing 97
playing in native players 92
shortcut for desktop 95
standalone 94
flipping
function curves 199
frame format
custom 74
default 73
playback 75
frame rates
custom 74
displaying in viewport 76
realtime playback 85
setting default 73
setting playback 75
frames
capturing in viewport 91
default format 73
Index

<table>
<thead>
<tr>
<th>Term</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>default rate</td>
<td>73</td>
</tr>
<tr>
<td>displaying number in viewport</td>
<td>80</td>
</tr>
<tr>
<td>start and end in timeline</td>
<td>78</td>
</tr>
<tr>
<td>FRAW (raw function curve) files</td>
<td>239</td>
</tr>
<tr>
<td>FRAW2 (raw function curve) files</td>
<td>about 239</td>
</tr>
<tr>
<td>converting expressions to</td>
<td>368</td>
</tr>
<tr>
<td>function curves</td>
<td></td>
</tr>
<tr>
<td>.fraw files</td>
<td>368</td>
</tr>
<tr>
<td>about 159</td>
<td></td>
</tr>
<tr>
<td>adding key points</td>
<td>195</td>
</tr>
<tr>
<td>animation editor</td>
<td>160</td>
</tr>
<tr>
<td>break point in slope</td>
<td>215</td>
</tr>
<tr>
<td>breaking and unifying slope</td>
<td>217</td>
</tr>
<tr>
<td>combining with expressions</td>
<td>367</td>
</tr>
<tr>
<td>converting between raw and standard</td>
<td>237</td>
</tr>
<tr>
<td>corresponding to action</td>
<td></td>
</tr>
<tr>
<td>sources 180</td>
<td></td>
</tr>
<tr>
<td>created from capturing with devices</td>
<td>381</td>
</tr>
<tr>
<td>cutting, copying, and pasting</td>
<td>206</td>
</tr>
<tr>
<td>cutting, copying, and pasting regions</td>
<td>207</td>
</tr>
<tr>
<td>cycles 223</td>
<td></td>
</tr>
<tr>
<td>default selection tool</td>
<td>187</td>
</tr>
<tr>
<td>default slope</td>
<td>216</td>
</tr>
<tr>
<td>default slope handle length</td>
<td>219</td>
</tr>
<tr>
<td>deleting key points</td>
<td>197</td>
</tr>
<tr>
<td>deselecting 186</td>
<td></td>
</tr>
<tr>
<td>displaying 173, 174</td>
<td></td>
</tr>
<tr>
<td>displaying in graph</td>
<td>175</td>
</tr>
<tr>
<td>displaying information about</td>
<td>177</td>
</tr>
<tr>
<td>displaying parameters in tree</td>
<td>174</td>
</tr>
<tr>
<td>exporting 239</td>
<td></td>
</tr>
<tr>
<td>extrapolating slope shape</td>
<td>222</td>
</tr>
<tr>
<td>fitting 234</td>
<td></td>
</tr>
<tr>
<td>flipping direction in X or Y</td>
<td>199</td>
</tr>
<tr>
<td>fraw and fraw2 files</td>
<td>239</td>
</tr>
<tr>
<td>from expressions</td>
<td>368</td>
</tr>
<tr>
<td>hiding 177</td>
<td></td>
</tr>
<tr>
<td>high-density (raw)</td>
<td>169</td>
</tr>
<tr>
<td>importing 239</td>
<td></td>
</tr>
<tr>
<td>increasing key points on</td>
<td>235</td>
</tr>
<tr>
<td>inserting key points without</td>
<td></td>
</tr>
<tr>
<td>changing curve shape</td>
<td>195</td>
</tr>
<tr>
<td>interpolation types</td>
<td>210</td>
</tr>
<tr>
<td>link curves</td>
<td>333</td>
</tr>
<tr>
<td>loading 239</td>
<td></td>
</tr>
<tr>
<td>locking on to an element</td>
<td>176</td>
</tr>
<tr>
<td>mirroring slope orientation</td>
<td>218</td>
</tr>
<tr>
<td>moving 191</td>
<td></td>
</tr>
<tr>
<td>muting (deactivating)</td>
<td>236</td>
</tr>
<tr>
<td>normalizing the view</td>
<td>178</td>
</tr>
<tr>
<td>offsetting 193</td>
<td></td>
</tr>
<tr>
<td>opening fcurve editor with specific</td>
<td>168</td>
</tr>
<tr>
<td>presets for non-time curves</td>
<td>238</td>
</tr>
<tr>
<td>quaternion rotation controls</td>
<td>226</td>
</tr>
<tr>
<td>redoing changes</td>
<td>168</td>
</tr>
<tr>
<td>reducing keyframes on</td>
<td>234</td>
</tr>
<tr>
<td>reducing noise</td>
<td>233</td>
</tr>
<tr>
<td>removing animation</td>
<td>198</td>
</tr>
<tr>
<td>removing flat (static) fcurves</td>
<td>198</td>
</tr>
<tr>
<td>resampling 235</td>
<td></td>
</tr>
<tr>
<td>rippling when pasting</td>
<td>207</td>
</tr>
<tr>
<td>saving 239</td>
<td></td>
</tr>
<tr>
<td>scaling 203</td>
<td></td>
</tr>
<tr>
<td>scaling regions</td>
<td>204</td>
</tr>
<tr>
<td>scaling with a pivot</td>
<td>205</td>
</tr>
<tr>
<td>selecting 185</td>
<td></td>
</tr>
<tr>
<td>selecting objects animated with</td>
<td>169</td>
</tr>
<tr>
<td>shaping high-density curves (HLE tool)</td>
<td>229</td>
</tr>
<tr>
<td>slope handles</td>
<td>212</td>
</tr>
<tr>
<td>slope orientation</td>
<td>214</td>
</tr>
<tr>
<td>smoothing 233</td>
<td></td>
</tr>
<tr>
<td>SOFTIMAGE</td>
<td>3D slope angle display</td>
</tr>
<tr>
<td>SOFTIMAGE</td>
<td>3D slope handles</td>
</tr>
<tr>
<td>spreadsheet query</td>
<td>169</td>
</tr>
<tr>
<td>temporary copies (snapshots)</td>
<td>184</td>
</tr>
<tr>
<td>undoing changes</td>
<td>168</td>
</tr>
<tr>
<td>See also animation editor, fcurve editor, key points, keyframes, slopes</td>
<td></td>
</tr>
<tr>
<td>functions in expressions</td>
<td>358</td>
</tr>
</tbody>
</table>

G

ghosting
- animated objects 42
- display types 49
- for groups and layers 43

gimbal lock
- Euler versus quaternion rotations 153

Global Transform property editor 141

global transforms
- animating 142
- automatically marking 145
- in expressions 366
- marking parameters for 145

graph editor See fcurve editor or animation editor

graphs
- function curve in animation editor 173
- viewing expressions as 352
- green box See animation icon
- grid
- fcurve graph 171
- groups
- null as parent, see transform groups 150
- transform (null parent) 150

H

has_relation function for expressions 366

hierarchies
- animating transformations in 148
- null parents (transform groups) 150

high-density function curves 169

editing 229

HLE tools (function curves) 229

I

icons
- animation 26

importing
- function curves 239
- scenes with expressions 366
Index

in-betweening 101
Info Scene
  animation 28
interpolation
  default for function curves 211
types for function curves 210
K
key points
  adding 195
cutting, copying, and pasting 206
deleting 197
deselecting 189
inserting 195
locking 200
moving 191
moving regions of 193
offsetting 193
scaling 203
selecting (tagging) 188
selecting by region 190
snapping 201
keyable parameter editor 111
keyable parameters 111
keyframe icon
  in a property editor 117
  in the Animation panel 107
keyframes
  about 101
  creating animation with 103
current values at different frames 135
in dopesheet 251
in-betweening 101
key points on function curves 188
moving between for parameters 123
on function curves (key points) 159
path animation from 277
preferences for keying 105
removing 69
removing in a property editor 119
removing transformations 147
save key command mapping 105
setting 103
setting automatically 121
setting for transformations 146
setting from a property editor 117
setting from the Animation panel 107
setting in keying panel 109
setting multiple at a time 108
setting quaternion rotations 155
workflow overview 103
keying panel
  changing parameter values 113
  muting animation in 116
  overview of setting keys 109
  setting keys 114
keys
  setting See keyframes
kinematics
  node in explorer 140
  properties of objects 141
L
l_fcv() expression for linked parameters 330
l_interp() expression for multi-linked parameters 337
l_interpOri expression for orientation linked parameters 341
line tests for animation 91
linear
  interpolation for function curves 210
linked parameters
  about 327
copying 342
  editing the falloff for multi-linked and orientation parameters 339
  editing the link function curve 333
l_fcv() expression 330
l_interp() expression 337
l_interpOri expression 341
linking a parameter to an object’s orientation 341
multiple parameters driving
  one 337
one-to-one 330
parameter connection editor 330
removing 342
setting relative values 332
viewing information 328
with color parameters 342
linking
  driving a parameter with an object’s orientation 341
  multiple parameters driving one parameter 337
  one parameter driving another 330
  parameters 327
loading
  expressions 365
  function curves 239
Local Transform property editor 141
local transforms
  animating 142
  automatically marking specific axes 145
  in expressions 366
  marking parameters for 144
local variables, in expressions 366
locking
  elements in animation tree 162
  key points 200
  marked parameters 38
  slope handles on function curves 221
looping
  animation playback 82
  quick loop 82
M
Make Preview  See capturing animation in a viewport
marked parameters
  automatically for global transforms 145
  automatically for specific local transform axes 145
  clearing (unmarking) 39
list 36
Index

locking 38
marking sets (remembering marked parameters) 39
moving between keys 123
plotting 61
setting 35
marking parameters
for animation 35
for transformations 144
in the dopesheet 250
marking sets
adding to or removing from 40
creating 39
deleting 40
selecting objects with 40
using 40
math operators
in expressions 359
mathematical expressions See expressions
memory sets See marking sets
metacurve region in fcurve editor 180
MIDI board (PC1600) device driver 375
MIDI keyboard device driver 375
mirroring
slope orientation on function curves 218
models
copying animation between using the dopesheet 259
motion capture data
cleaning up function curves 234
editing function curves from 229
motion guides See paths
motion path action See paths
motion trails 49
mouse
device driver for 375
multiple-linked parameters 337
multiple-point constraints 305
muting
animation 65
animation in dopesheet 261
animation in keying panel 116
constraints 324
expressions 364
function curves 236
N
noise
reducing on function curves 233
normals
constraining object to 308
null parent object, see transform groups 150
O
offsetting
animation 55, 193
between constrained objects 318
onion-skinning See ghosting operators
math in expressions 359
orientation
constraints 293
linking to a parameter 341
oscillation
in expressions 372
panels
Animation 25
Playback 80
panning
in dopesheet 245
in fcurve editor 182
parameter connection editor 330
parameter curves See function curves
parameters
in expressions 358
linking 327
marked 35
setting keys for 117
path animation 265
controlling flipping for up vector 315
paths
about 265
constraining objects to curve 272
converting object translation to 278
creating from keys 277
deleting 282
editing the curve 281
editing the timing 279
elements in explorer 267
locking keys on 281
percentage function curve 279
removing 282
setting objects on 268
substituting curve 281
tangency 269
trajectories 274
viewing information 266
pencil tests 91
pinning
constraining object in place 292
pivot point 152
planting See pinning
playback
controls 80
cursor in animation editor 164
cursor in timeline 79
default frame rate 73
frame format 75
frame rate 75
looping animation 82
optimizing 89
realtime 85
realtime in flipbooks 94
scrubbing 83
setting range of frames 77
setting start and end frames 78
Playback panel 80
playblast See capturing animation in a viewport 91
playing
animation 77, 80
flipbooks 97
plotting
animation 59
curves 274
Index

Animation 389

pose constraints 300
position constraints 291
preferences
  animation 28
  animation tree in animation editor 163
dopesheet 243
fcurve editor 166
timeline keys 138
preferred angles
  and expressions 374
presets
  for expressions 365
  for non-time function curves 238
property editors
  keyframing parameter values in 117
Q
quaternion rotations
  about 153
  converting to and from Euler 154
  editing function curves 226
  keying 155
quaternions
  linking a parameter to an object’s orientation 341
queries
  animation in spreadsheet 32
R
range of frames for timeline 77
raw function curves 169
  converting to standard 237
  editing 229
realtime playback
  about 85
  in flipbooks 94
  rate percentage 85
  with fast playback 86
regions
  of function curves 190
  on dopesheet tracks 252
relational expressions 369
relative values
  for linked parameters 332
relaxing constraints
  See removing, constraints
removing
  animation 69
  animation from function curves 198
  constraints 325
  expressions 361
  flat (static) function curves 198
  keyframes in a property editor 119
  link animation 342
  transformation keys 147
  up-vector constraints 315
resampling
  function curves 235
reversing
  animation 56
ripple option
  in dopesheet 127, 253
  in fcurve editor 207
roll-frame normal for path 315
rotations
  animating 139
  constraints, See constraints, orientation
  continuous, fcurves 156
  continuous, plotting 64
  plotting 61
  quaternion 153
S
saving
  expressions 365
  function curves 239
scaling
  animation 55
  constraints 299
schematic view
  animation elements in 31
scripted operators
  in animation editor 162
scrubbing
  modes and playback rates 83
playback cursor 79
selecting
  animated objects 29
  constrained objects 287
  constraining object 287
Sequence commands
  scaling and offsetting
  animation 55
SI_TXT_EDITOR_ENV 360
slope handles
  angle and length 212
  equal length 218
  fixed length (SOFTIMAGE|3D style) 219
  locking 221
  on function curves 212
  snapping 220
slopes
  of function curves 212
smoothing
  function curves 233
snapping
  key points on function curves 201
  slope handles on function curves 220
snapshots of function curves for editing 184
SOFTIMAGE|3D
  slope angle display for function curves 214
  slope handle style for function curves 219
spline
  interpolation for function curves 210
spreadsheet
  animation queries in 32
  standalone flipbook (flip.exe) 94
  start frame in timeline 78
static channels
  removing animation from 70
  static channels, deleting 198
stepped keys (constant fcurve interpolation) 211
surfaces
Index

constraining objects to 308
symmetry
constraint 301

T
Tag2Path 274
tangency
on curves 269
tangent handles See slope handles, function curves
this, token in expressions 357
this_model, token in expressions 357
three-point constraints 304
time
in scene 71
See also timeline
time range 77
time-display format
animation editor 171
timeline
about 79
in animation editor 164
keying current values at different frames 135
looping playback 82
mini dopesheet view 243
playback cursor 79
range of frames 77
setting preferences for keys in 138
start and end frames 78
time format 78
toolbars
Animate 24
Refer also to Online Help
tracks
in dopesheet 247
trails for animated objects 49
trajectories
animation on (path) 274
transform groups for parenting 150
transformations
animating 139
animating centers 152
animating hierarchies 148
keyframing 146
kinematics property editor 141
marking parameters for 144
plotting 61
removing keys 147
translations
animating 139
creating path animation from 278
plotting (position) 61
tree (explorer) in animation editor 162
two-point constraints 303
U
up vector
controlling flipping with path animation 315
up vectors
constraints for objects 312
removing constraints 315
V
viewing
captured animation 94
constraint information 288
expression information 346
flipbooks 94
optimizing playback for animation 89
path information 266
viewports
capturing animation in 91
displaying frame number in 80
displaying frame rate in 76
W
walk cycles
preventing backsliding with plateau fcurve slopes 216
Z
zooming
in dopesheet 245
in fcurve editor 181, 182